

NW-11

WATCHDOG DIAGNOSTIC
MD-11-DZNSWA-A

EP-DZNSWA-A-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames, each displaying diagnostic data for the MD-11 aircraft. The data is organized into columns and rows, with some frames containing text and others containing graphical representations like bar charts or histograms. The frames are arranged in a regular grid pattern, typical of microfiche format.

.REM 8

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZMWA-A
PRODUCT NAME: MW11 WATCHDOG DIAGNOSTIC
DATE: 4-FEB-76
MAINTENANCE: DIAGNOSTIC GROUP
AUTHOR: PAUL NELSON

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION.

TABLE OF CONTENTS

1. ABSTRACT
2. REQUIREMENTS (EQUIPMENT & MEMORY)
3. LOADING PROCEDURE
4. STARTING PROCEDURE
5. OPERATING PROCEDURE
6. ERRORS - GENERAL
7. RESTRICTIONS
8. MISCELLANEOUS
9. PROGRAM TESTS DESCRIPTION

NOEC-11-DJMA-A
SAC

130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

2. REQUIREMENTS

2.1 EQUIPMENT

A PDP 11 FAMILY COMPUTER WITH AT LEAST 8K OF MEMORY, A CONSOLE AND A KW11-L CLOCK OPTION ALONG WITH THE NW11 A OR C.

3. LOADING PROCEDURE

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

STANDARD PDP 11 FORMAT

- SW15 = 1 HALT ON ERROR.
- SW14 = 1 LOOP ON TEST
- SW13 = 1 INHIBIT ERROR TYPEOUTS
- SW11 = 1 INHIBIT ITERATIONS
- SW10 = 1 LOOP ON CYCLIC TESTS
(TESTS 3 THRU 26)
- SW9 = 1 LOOP ON ERROR
- SW8 = 1 LOOP ON TEST IN SWR<7:0>

SPECIAL NOTE

IF THE COMPUTER UTILIZED IS A LSI 11 OR A COMPUTER WITHOUT SWITCH REGISTER. THE PROGRAM WILL UTILIZE LOCATIONS 174 AND 176 AS A "DISPLAY" REGISTER AND A "SWITCH" REGISTER RESPECTIVELY. THE OPERATOR WILL BE RESPONSIBLE FOR THE LOADING OF THE "SWITCH" REGISTER LOCATION PRIOR TO STARTING OR RESTARTING THE PROGRAM.

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
1874.2 STARTING ADDRESSES

200 IS THE INITIAL OR NORMAL STARTING ADDRESS
204 IS THE RESTART ONLY ADDRESS

5.0 OPERATING PROCEDURE

5.1 FULL LOGIC TEST (SA-200).

IN THIS MODE THE DIAGNOSTIC WILL VERIFY THE OPERATION OF THE SYSTEM'S NW11-A, THE ABILITY TO PROPERLY ADDRESS THE UNIT WITHOUT TRAPPING, AND OTHER FUNCTIONS WHICH REQUIRE MANUAL INTERVENTION. INITIALLY TESTING MUST BEGIN WITH A START FROM LOCATION 200, SINCE ALL REGISTER ADDRESSES AND VECTORS ARE SET UP IN THE INITIALIZING SECTION.

5.2 RESTART (SA-204)

STARTING FROM THIS POINT ALLOWS THE OPERATOR TO BEGIN TESTING USING THE PREVIOUSLY ENTERED ADDRESS AND VECTOR.

5.3 TESTING PROCEDURE

INITIALLY THE TEST SHOULD BE RUN FROM A S.A. OF 200 WITH ALL SWITCHES DOWN. THIS WILL CHECK FOR ALL POSSIBLE LOGICAL FAILURES WHICH CAN BE DETECTED VIA THE NW11'S REGISTERS AND ALSO PROVIDES SCOPING LOOPS FOR TIMING FUNCTIONS WHICH ARE EITHER NOT ON THE BUS OR CAN BE VARIED VIA SWITCHES ON NW11 ITSELF.

AFTER AT LEAST ONE PASS OF THE COMPLETE TEST HAS BEEN SUCCESSFULLY EXECUTED, SW 10 MAY BE RAISED (SET TO A ONE) TO ALLOW THE TEST TO LOOP THROUGH TESTS 3 THRU 26 WITHOUT ANY FURTHER OPERATOR INTERVENTION.

188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240

5.4 SPECIAL TEST EXITS (TESTS 27 TO 36 ONLY)

TESTS 27 THRU 36, BEING DEPENDENT UPON THE CORRECT OPERATION OF THE PANEL SWITCHES FOR NORMAL TEST EXITS, HAVE BEEN PROVIDED WITH AN ALTERNATIVE TEST EXIT WHICH IS THE CONTROL C (OCTAL 003) FUNCTION FROM THE CONSOLE KEYBOARD. STRIKING A CONTROL C WILL SIMULATE PROPER PANEL RESPONSES (IE. NO ERRORS WILL BE REPORTED). CONTROL C IS THE ONLY METHOD OF EXITING TESTS 30, 34, 35, AND 36.

6. ERRORS

6.0 GENERAL

ALMOST ALL ERRORS PROVIDE SPECIFIC FAILURE INFORMATION AND ARE GENERALLY ASSOCIATED WITH ONLY ONE TESTS FAILURES. THE STANDARD ERROR WILL CONTAIN THE FOLLOWING INFORMATION:

- A. AN ERROR MESSAGE OF THE SPECIFIC FAILURE SITUATION
- B. ERROR PC - THE PROGRAM LOCATION OF THE ERROR.
- C. REG. - THE ADDRESS OF THE DISPLAYED REGISTER
- D. GOOD - WHAT THE REGISTER SHOULD HAVE CONTAINED.
- E. BAD - THE ACTUAL CONTENTS OF THE REGISTER AT FAILURE TIME.

NOTE:

IN SOME INSTANCES THE "GOOD" DISPLAY MAY CONTAIN OR LACK SEVERAL DIFFERENCES FROM THE "BAD" DISPLAY. IN MANY CASES THIS WILL BE DUE TO THE "TOGGLING" OF TIME-3 AT ONE SECOND INTERVALS AND THE "TIMING OUT" OF TIME-1 AND TIME-2. THE ERROR MESSAGE IN THESE CASES, WILL INDICATE WHICH SIGNAL (OR BIT) ACTUALLY RESULTED IN THE FAILURE.

2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795

7. RESTRICTIONS

- A. INITIALLY THE TEST MUST BE STARTED FROM LOCATION 200.
- B. INITIAL SETUP MESSAGE MUST BE FOLLOWED TO AVOID EXTRANEIOUS BOOTS AND OTHER ERRORS.
- C. IT WOULD BE DESIRABLE, ALTHOUGH NOT ABSOLUTELY NECESSARY TO DISCONNECT THE "BOOT" SIGNAL LINES TO INSURE A HARDWARE FAILURE DID NOT INADVERTANTLY RESULT IS A "BOOT" SIGNAL WHICH MAY OVERLAY THE DIAGNOSTIC.
- D. NW11 ADDRESSES MUST FALL WITHIN THE "FLOATING ADDRESSES" BETWEEN 760010 TO 763776.

8. MISCELLANEOUS

- A. IF THE NW11 ADDRESS THAT IS ENTERED ENTERED RESULTS IN A "TRAP" MESSAGE RECHECK THE ADDRESS OF THE BOARD, PRESS CONTINUE, AND ENTER THE PROPER ADDRESS.

9. PROGRAM DESCRIPTION

9.0 INITIALIZATION

THIS SEGMENT OF THE TEST PERFORMS ALL STANDARD SYSMAC INITIALIZATION, THEN ASKS THE OPERATOR FOR THE ADDRESS OF THE NW11'S REGISTER 0. THIS IS CHECKED TO INSURE A TRAP WILL NOT RESULT WHEN THE NW11 IS ADDRESSED. THE PRESENCE AND OPERATION OF A KW11 CLOCK IS VERIFIED AND ANY PROBLEMS REPORTED.

9.1 TEST 1 INSURE VECTOR CAN BE READ

THIS ROUTINE WILL ATTEMPT TO "READ" REGISTER 2 OF THE NW11A. IF THE "READ" RESULTS IN A "TRAP" A TRAP MESSAGE WILL BE DISPLAYED AND THE PROGRAM WILL HALT. PRESSING CONTINUE WILL CAUSE THE PROGRAM TO RETURN TO THE ADDRESS ENTRY SECTION OF THE INITIALIZATION ROUTINE AND WAIT FOR INPUT FROM KEYBOARD. IF THE "READ" WAS SUCCESSFUL THE ADDRESS READ WILL BE DISPLAYED ON THE CONSOLE.

IF THE SIGNAL BUS DATA L IN THE NW11 ALWAYS LOW, THE PROGRAM WILL HALT AT THE NW11 VECTOR + 2.

296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350

9.2 TEST 2 CHECK STATE OF NW11 AFTER POWER RESET.

ROUTINE TO INSURE STATIC INTEGRITY OF THE NW11 AFTER THE OPERATOR HAS POWERED THE HOST DOWN THEN UP. REGISTER 0 WILL HAVE BIT 10 SET AND BIT 3 SETTING AND RESETTING. REGISTER 2 IS THE VECTOR AND IS NOT CHECKED IN THIS TEST. REGISTERS 4 AND 6 SHOULD CONTAIN ALL ZEROS.

A STATUS REGISTER ERROR (ERROR 1) IS ISSUED IF ANY NON-STANDARD CONDITIONS ARE ENCOUNTERED.

9.3 TEST 3 INSURE READ/WRITE BITS CAN BE SET

ROUTINE TO INSURE ALL READ/WRITE BITS IN REGISTER 0, 4, AND 6 CAN BE SET. OCTAL EQUIVALENTS OF THE BITS WHICH ARE INDEPENDENTLY SET AND RESET ARE:
REG.0 = 104, REG.4 = 177000, AND REG.6 = 177570

A STATUS REGISTER ERROR (ERROR 1) IS ISSUED IF ANY BIT CANNOT BE WRITTEN, THEN READ.

9.4 TEST 4 INSURE TIME 1 AND 2 (REG. 0, BITS 8-9) SET AND RESET, INSURE TIME 2 EXCEEDS TIME 1.

SUBTEST A ISSUES A TGO (REG.0, BIT 0) CAUSES BOTH T1 AND T2 TO SET THEN ALLOWS 3 SECONDS FOR BOTH TO RESET.

ERROR 2 IS ISSUED IF ONE OR BOTH DO NOT SET AND ERROR 3 IS ISSUED IF BOTH DO NOT RESET.

SUBTEST B ISSUES TGO AND INSURES THAT TIME 1 RESETS BEFORE TIME 2.

ERROR 4 IS ISSUED IF T2 RESETS BEFORE T1.

9.5 TEST 5 CHECK THAT ALARM (REG0, BIT7) CAN BE SET AND RESET.

ROUTINE ISSUES TGO AND WAITS FOR T-2 TO RESET. AT THAT TIME ALARM ON THE BUS SHOULD BE SET

ERROR 5 IS ISSUED IF ALARM DOES NOT SET.

ROUTINE THEN ISSUES A SERIES OF 5 TGO'S AND INSURES THAT THE CLEAR 1 PULSE IN THE NW11 CLEARS ALARM.

ERROR 6 IS ISSUED IS THE ALARM WAS NOT RESET.

351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403

9.6 TEST 6 CHECK THAT WDE(REG.0,BIT1) CAN BE SET IN MAINTENANCE MODE.

ROUTINE SETS THE MAINT BIT (REG.0,BIT2) THEN CHECKS TO SEE THAT WDE CAN BE SET AND READ. ROUTINE THEN CLEARS MAINT AND INSURES THAT WDE CANNOT BE SET AND READ.

ERROR 7 IS ISSUED IF WDE CANNOT BE SET AND ERROR 7 IS ISSUED IF IT CAN BE SET WITHOUT "MAINT" BEING SET.

9.7 TEST 7 INSURE SETTING DROUT 1 (REG.6,BIT8) FORCES RECEIVER IN (REG.0,BIT15).

TEST TO VERIFY THAT SETTING DROUT 1 WILL VIA THE MAINTENANCE SWITCH, SET RECEIVER IN. TEST INSURES REC.IN. SETS THEN RESETS WITHIN 32 M.S;

ERROR 11 IS ISSUED IF REC.IN CANNOT BE SET OR WILL NOT RESET.

9.10 TEST 10 CHECK BOOT ENABLE (REG.0,BIT4) SET AND RESET CONDITIONS.

ROUTINE INSURES THAT BOOT ENABLE CAN BE SET BY AN AND OF MAINT + BTENA AND WILL ALSO BE SET BY RECV. IN.

TEST THEN INSURES THAT GENERATING A CLEAR 1 WILL RESET BOOT ENABLE.

ERROR 13 IS ISSUED IF BOOT ENABLE CANNOT BE SET AND ERROR 12 IS ISSUED IF IT CANNOT BE RESET.

9.11 TEST 11 CHECK THAT GREEN LITE (REG.0,BIT12) AND AMBER LITE (REG.0 BIT13) CAN BE SET.

ROUTINE SUPPORTS THE NW11 T1-T2 SEQUENCE FOR A TOTAL OF 5-TGOS AFTER WHICH GLITE AND AMLITE ARE VERIFIED TO BE SET. T-2 IS THEN ALLOWED TO TIME OUT AND GLITE AND AMLITE ARE VERIFIED TO BE EXTINGUISHED.

ERROR 14 IS ISSUED IF EITHER GLITE OR AMLITE DO NOT SET AND ERROR 15 IS ISSUED IF EITHER DO NOT RESET.

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
456
457

9.12 TEST 12 CHECK DURATION OF TIME 3

ROUTINE INSURES THAT TM3 (REG.0,BITS) WILL BOTH SET AND RESET AND THAT THE DURATION IS 1 SECOND PLUS OR MINUS 10%.

ERROR 16 IS ISSUED IF TM3 WILL NOT SET OR RESET AND ERROR 17 IS ISSUED IF THE CALCULATED TIME IS NOT WITHIN SPECS.

9.13 TEST 13 CHECK PANEL SWITCHES (REG4,BITS0-7)

ROUTINE "READS" REG. 4 AND CHECKS TO INSURE THAT THE "ENTER SWITCHLOCK" ARE "OFF".

ERROR 20 IS ISSUED IF ANY SWITCH OR CORRESPONDING BIT IS SET.

9.14 TEST 14 CHECK THAT DROUT FLOPS (REG.4,BITS 8-15) CAN BE SET AND RESET

ROUTINE INSURES THAT BY WRITING BITS 8-15 TO REG 4, ALL DROUT FLOPS SET AND THAT THE INTERNAL 16.M.S. ONE-SHOT CLEARS THEM.

ERROR 21 IS ISSUED FOR BOTH SET AND RESET FAILURES.

9.15 TEST 15 CHECK FRO FLOP (REG 6, BIT 1) WILL SET AND RESET

ROUTINE INSURES FRO FLOP CAN BE SET BY SETTING WDE AND AN ALARM. TEST THEN INSURES THAT THE FRO FLOP CAN BE RESET VIA THE RESETTING OF WDE AND ISSUING A RESET COMMAND.

ERROR 22 IS ISSUED IF THE FRO FLOP CANNOT BE SET AND RESET PROPERLY.

9.16 TEST 16 CHECK MINUTE FLOPS (REG.6, BITS3-6)

ROUTINE TO INSURE MINUTE FLOPS (1,2,4 AND 8) CAN BE DIRECTLY SET IN MAINTENANCE MODE VIA THE APPROPRIATE MINUTE WRITE BIT. TEST THEN INSURES GENERATING A CLR 1 CLEARS ALL MINUTE FLOPS.

ERROR 23 IS ISSUED IF ANY MINUTE FLOPS CANNOT BE SET OR RESET UPON COMMAND.

458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

9.17 TEST 17 CHECK THAT 15 SEC. FLOP (REG.6,BIT1) CAN BE SET AND RESET.

ROUTINE TO INSURE THAT GENERATING 16 "SECONDS" IN MAINTENANCE MODE WILL SET THE 15 SEC FLOP AND THEN GENERATING 15 ADDITIONAL "SECONDS" WILL RESET THE 15 SEC FLOP.

ERROR 24 IS ISSUED IF THE 15 SEC FLOP WILL NOT SET AND ERROR 25 IS ISSUED IF IT WILL NOT RESET.

9.20 TEST 20 CHECK THAT 30 SEC. FLOP (REG.6,BIT2) CAN BE SET AND RESET.

ROUTINE TO INSURE THAT GENERATING 31 "SECONDS" IN MAINTENANCE MODE WILL SET THE 30 SEC. FLOP AND THEN GENERATING AN ADDITIONAL 30 "SECONDS" WILL RESET THE FLOP.

ERROR 26 IS ISSUED IF THE 30 SEC FLOP CANNOT BE SET AND ERROR 27 IS ISSUED IF IT CANNOT BE RESET.

9.21 TEST 21 CHECK THAT CLR 1 WILL RESET 15 AND 30 SEC. FLOPS.

ROUTINE SETS 15 AND 30 SEC FLOPS AS PER 9.17 AND 9.20 THEN GENERATES A CLR 1 AND INSURES BOTH FLOPS ARE CLEARED.

ERROR 30 IS ISSUED IF EITHER FLOP WILL NOT CLEAR.

9.22 TEST 22 INSURE PROPER SERIAL PROPOGATION FROM 15 SECONDS TO 15 MINUTES

ROUTINE GENERATES A SERIES OF 15 "SECONDS" AND INSURES THAT ALL FLOP CONFIGURATIONS CAN BE GENERATED SERIALY. A "RE-BOOT" IS EXPECTED TO OCCUR ONCE DURING THIS TEST AND THE TIME CONFIGURATION (MIN. SEC.) WHICH CAUSED THE REBOOT IS REPORTED.

ERROR 33 IS ISSUED IF ANY ERRONEAOUS TIME PROPOGATION IS DISCOVERED.

9.23 TEST 23 CHECK THAT A TRANSITION OF THE 1 SECOND FLOP (TIME3;REG.0, BITS) WILL TOGGLE THE 15 SECOND COUNTER.

ROUTINE GENERATES 15 "SECONDS" AND THEN WAITS FOR THE TIME 3 CLOCK TO TOGGLE THE 15 SEC COUNTER AND SET THE 15 SEC.FLOP.

ERROR 31 IS ISSUED IF THE 15 SEC FLOP IS NOT SET WITHIN 3

MO1

MAINDEC-11-DZWA-A
DZWA.SRC

MACY11 27(732) 25-SEP-76 12:05 PAGE 13

514

ACTUAL SECONDS.

515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570

9.24 TEST 24 CHECK BOOT GENERATION CONDITIONS

ROUTINE INSURE THAT A "BOOT" WILL BE GENERATED BY SETTING "BTENA" IN MAINTENANCE MODE AND BY GENERATING THE MINUTE-SECOND CONFIGURATION REQUIRED TO GENERATE A "BOOT".

ERROR 32 WILL BE ISSUED IF A "BOOT" CANNOT BE GENERATED AND ERROR 36 WILL BE ISSUED IF THE BOOT ONE-SHOT DOES NOT RESET WITHIN 2 ACTUAL SECONDS.

9.25 TEST 25 CHECK THE TIME 1 INTERRUPT.

ROUTINE SETS "T1INT" (REG.4,BIT9) AND MAINTENANCE MODE AND ISSUES A TGO TO THE NW11. IF A "T1-NOT" INTERRUPT IS NOT GENERATED WITHIN 48.M.S., AN ERROR IS ISSUED.

ERROR 37 IS ISSUED IF THE INTERRUPT DOES NOT OCCUR.

9.26 TEST 26 CHECK T1-T2 SEQUENCE.

ROUTINE GENERATES 5 TGOs AND INSURE THAT ALARM IS EXTINGUISHED AND THE GREEN LITE IS ON. T1 SUPPORT IS THEN DROPPED BY ISSUING A TGO BEFORE THE PREVIOUS T-1 HAS TIMED OUT AND AN ALARM IS EXPECTED. T-1 SUPPORT IS THEN DROPPED BY ALLOWING IT TO TIME OUT AND THE SAME RESULTS ARE EXPECTED.

ERROR 6 IS ISSUED IF ALARM CANNOT BE CLEARED, ERROR 34 IS ISSUED IF DROPPING T-1 SUPPORT DOES NOT CAUSE ON ALARM AND ERROR 5 IS ISSUED IF DROPPING T-2 SUPPORT DOES NOT CAUSE AN ALARM.

*** END OF CYCLIC TESTS ***
*** FOLLOWING ARE INTERVENTION TESTS ***

9.27 TEST 27 CHECK "SWITCH LOCK" ENABLE (REG4,BIT0)

INSURE PANEL SWITCHES CAN BE ENABLED VIA "SWITCHLOCK" SWITCH.

9.30 TEST 30 CHECK PANEL SWITCHES

PANEL SWITCHES ARE LOOPED INTO PANEL LAMPS FOR VISUAL VERIFICATION.

610
611
612
613
614
615
616

9.31 TEST 31 CHECK "ENTER" SWITCH CAUSES INTERRUPT.

ROUTINE INSURES THAT LIFTING THE "ENTER" SWITCH ONCE CAUSES ONE AND ONLY ONE INTERRUPT. NO INTERRUPT AND MULTIPLE INTERRUPTS ARE REPORTED.

9.32 TEST 32 CHECK "ALARM/LOAD" SWITCH -A.

ROUTINE INSURES TURNING THE "ALARM/LOAD" SWITCH TO THE "ALARM ENABLE" POSITION SETS WDE (REG.D,BIT1) AND CONVERSLY IT CLEARS WHEN THE SWITCH IS RETURNED TO THE "DISABLE" POSITION.

9.33 TEST 33 CHECK "ALARM/LOAD" SWITCH -B

ROUTINE TO INSURE THAT WHEN THE "ALARM/LOAD" SWITCH IS TURNED TO "LOAD ENABLE" AND "LOAD" SWITCH IS LIFTED THE BOOT ENABLE FLOP IS SET. ROUTINE THEN INSURES THAT BOOT ENABLE FLOP CAN BE RESET VIA A CLR 1 PULSE AFTER THE "LOAD" SWITCH IS RELEASED.

9.34 TEST 34 SCOPE LOOP FOR FIELD RELAY.

ROUTINE TOGGLES THE FRO FLOP AT 5 SECOND INTERVALS. C/R EXITS LOOP.

9.35 TEST 35 SCOPE LOOP FOR DROUT FLOPS.

ROUTINE TOGGLES DROUT FLOPS SEQUENTIALLY AT ONE SECOND INTERVALS. C/R EXITS LOOP.

9.36 TEST 36 T-1, T-2 SCOPE LOOPS

ROUTINE SETS UP TIME 1 AND TIME 2 SCOPING LOOPS, ONE AT A TIME. C/R EXITS EACH SUBTEST LOOP.

673
674
675
676
677 000000
678 000040
679 000100
680 000140
681 000200
682 000240
683 000300
684 000340
685
686
687 100000
688 040000
689 020000
690 010000
691 004000
692 002000
693 001000
694 000400
695 000200
696 000100
697 000040
698 000020
699 000010
700 000004
701 000002
702 000001
703
704
705
706
707
708
709
710
711
712
713
714
715 100000
716 040000
717 020000
718 010000
719 004000
720 002000
721 001000
722 000400
723 000200
724 000100
725 000040
726 000020
727 000010
728 000004

.EQUIV R6,SP ;:STACK POINTER
.EQUIV R7,PC ;:PROGRAM COUNTER

.*PRIORITY LEVEL DEFINITIONS
PR0= 0 ;:PRIORITY LEVEL 0
PR1= 40 ;:PRIORITY LEVEL 1
PR2= 100 ;:PRIORITY LEVEL 2
PR3= 140 ;:PRIORITY LEVEL 3
PR4= 200 ;:PRIORITY LEVEL 4
PR5= 240 ;:PRIORITY LEVEL 5
PR6= 300 ;:PRIORITY LEVEL 6
PR7= 340 ;:PRIORITY LEVEL 7

.*"SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4

```

729      000002      BIT01= 2
730      000001      BIT00= 1
731      .EQUIV      BIT09,BIT9
732      .EQUIV      BIT08,BIT8
733      .EQUIV      BIT07,BIT7
734      .EQUIV      BIT06,BIT6
735      .EQUIV      BIT05,BIT5
736      .EQUIV      BIT04,BIT4
737      .EQUIV      BIT03,BIT3
738      .EQUIV      BIT02,BIT2
739      .EQUIV      BIT01,BIT1
740      .EQUIV      BIT00,BIT0
741
742      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
743      000004      ERRVEC= 4          ;: TIME OUT AND OTHER ERRORS
744      000010      RESVEC= 10       ;: RESERVED AND ILLEGAL INSTRUCTIONS
745      000014      TBITVEC=14       ;: "T" BIT
746      000014      TRTVEC= 14       ;: TRACE TRAP
747      000014      BPTVEC= 14       ;: BREAKPOINT TRAP (BPT)
748      000020      IOTVEC= 20       ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
749      000024      PWRVEC= 24       ;: POWER FAIL
750      000030      EMTVEC= 30       ;: EMULATOR TRAP (EMT) **ERROR**
751      000034      TRAPVEC=34       ;: "TRAP" TRAP
752      000060      TKVEC= 60        ;: TTY KEYBOARD VECTOR
753      000064      TPVEC= 64        ;: TTY PRINTER VECTOR
754      000240      PIRQVEC=240      ;: PROGRAM INTERRUPT REQUEST VECTOR
755      .SBTTL      TRAP CATCHER
756
757      000000      .=0
758      ;*ALL UNUSE(7) LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
759      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
760      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
761      000174      000174      .=174
762      000174      000000      DISPREG: .WORD 0          ;: SOFTWARE DISPLAY REGISTER
763      000176      000000      SWREG:   .WORD 0          ;: SOFTWARE SWITCH REGISTER
764      .SBTTL      ACT11 HOOKS
765
766      ;:*****
767      ;:HOOKS REQUIRED BY ACT11
768      000200      $SVPC=.          ;:SAVE PC
769      000046      .=46
770      000046      011352      SENDAD          ;:1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
771      000052      000052      .=52
772      000052      000000      .WORD 0          ;:2)SET LOC.52 TO ZERO
773      000200      000200      .=$SVPC          ;: RESTORE PC
774      000200      000200      .=200
775      000200      000137      001746      START: JMP      NS"ART          ;:INITIAL START
776      000204      000204      .=204
777      000204      000137      001736      RSTART: JMP      LSTART          ;:RESTART TEST

```

.SBTTL COMMON TAGS

778
779
780
781
782
783
784 001100
785 001100
786 001100 000000
787 001100 000
788 001103 000
789 001104 000000
790 001106 000000
791 001110 000000
792 001112 000000
793 001114 000
794 001115 001
795 001116 000000
796 001120 000000
797 001122 000000
798 001124 000000
799 001126 000000
800 001130 000000
801 001132 000000
802 001134 000
803 001135 000
804 001136 000000
805 001140 177570
806 001142 177570
807 001144 177560
808 001146 177562
809 001150 177564
810 001152 177566
811 001154 000
812 001155 002
813 001156 012
814 001157 000
815 001160 000000
816 001162 000000
817 001164 077
818 001165 015
819 001166 000012
820

```

*****
; THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
; USED IN THE PROGRAM.

```

```

.=1100
$CMTAG:          .WORD      0          ; START OF COMMON TAGS
$PASS:          .WORD      0          ; CONTAINS PASS COUNT
$STNM:          .BYTE      0          ; CONTAINS THE TEST NUMBER
$ERFLG:         .BYTE      0          ; CONTAINS ERROR FLAG
$ICNT:          .WORD      0          ; CONTAINS SUBTEST ITERATION COUNT
$LPADR:         .WORD      0          ; CONTAINS SCOPE LOOP ADDRESS
$LPERR:         .WORD      0          ; CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL:         .WORD      0          ; CONTAINS TOTAL ERRORS DETECTED
$ITEMB:         .BYTE      0          ; CONTAINS ITEM CONTROL BYTE
$ERMAX:         .BYTE      1          ; CONTAINS MAX. ERRORS PER TEST
$ERRPC:         .WORD      0          ; CONTAINS PC OF LAST ERROR INSTRUCTION
$GADR:          .WORD      0          ; CONTAINS ADDRESS OF 'GOOD' DATA
$BADR:          .WORD      0          ; CONTAINS ADDRESS OF 'BAD' DATA
$GDAT:          .WORD      0          ; CONTAINS 'GOOD' DATA
$BDAT:          .WORD      0          ; CONTAINS 'BAD' DATA
                .WORD      0          ; RESERVED--NOT TO BE USED
                .WORD      0          ;
$AUTOB:         .BYTE      0          ; AUTOMATIC MODE INDICATOR
$INTAG:         .BYTE      0          ; INTERRUPT MODE INDICATOR
                .WORD      0          ;
$SWR:           .WORD      DSWR       ; ADDRESS OF SWITCH REGISTER
$DISPLAY:       .WORD      DDISP      ; ADDRESS OF DISPLAY REGISTER
$TKS:           177560                ; TTY KBD STATUS
$TKB:           177562                ; TTY KBD BUFFER
$TPS:           177564                ; TTY PRINTER STATUS REG. ADDRESS
$TPB:           177566                ; TTY PRINTER BUFFER REG. ADDRESS
$NULL:          .BYTE      0          ; CONTAINS NULL CHARACTER FOR FILLS
$FILLS:         .BYTE      2          ; CONTAINS # OF FILLER CHARACTERS REQUIRED
$FILLC:         .BYTE      12         ; INSERT FILL CHARS. AFTER A "LINE FEED"
$TPFLG:         .BYTE      0          ; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
$TIMES:         0                    ; MAX. NUMBER OF ITERATIONS
$ESCAPE:        0                    ; ESCAPE ON ERROR ADDRESS
$QUES:          .ASCII  '?'          ; QUESTION MARK
$CRLF:          .ASCII  <15>         ; CARRIAGE RETURN
$LF:            .ASCIIZ <12>         ; LINE FEED
*****

```

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
 ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 ;*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
 ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;* EM ;;POINTS TO THE ERROR MESSAGE
 ;* DH ;;POINTS TO THE DATA HEADER
 ;* DT ;;POINTS TO THE DATA
 ;* DF ;;POINTS TO THE DATA FORMAT

\$ERRTB:

;STATUS REGISTER ERROR

821
822
823
824
825
826
827
828
829
830
831
832
833
834
835 001170
836
837
838
839 001170 015434
840 001172 015525
841 001174 001570
842 001176 001602
843
844
845
846
847 001200 015464
848 001202 015525
849 001204 001570
850 001206 001602
851
852
853
854 001210 015564
855 001212 015525
856 001214 001570
857 001216 001602
858
859
860
861 001220 015644
862 001222 015525
863 001224 001570
864 001226 001602
865
866
867
868 001230 015704
869 001232 015525
870 001234 001570
871 001236 001602
872
873
874
875 001240 015764
876 001242 015525

;T1 OR T2 DID NOT SET

EM2
DH1
DT1
DF1

;T1 OR T2 FAILED TO RESET WITHIN 3 SECONDS

EM3
DH1
DT1
DF1

;T1 EXCEEDS T2 ERROR

EM4
DH1
DT1
DF1

;NO ALARM FROM T2 TIMEOUT

EM5
DH1
DT1
DF1

;SUPPORTING T1-T2 DID NOT CLEAR ALARM

EM6
DH1

| | | | |
|-----|--------|--------|-----------------------------------|
| 877 | 001244 | 001570 | DT1 |
| 878 | 001246 | 001602 | DF1 |
| 879 | | | |
| 880 | | | ;WDE WILL NOT SET IN MAINT MODE |
| 881 | | | |
| 882 | 001250 | 016055 | EM7 |
| 883 | 001252 | 015525 | DH1 |
| 884 | 001254 | 001570 | DT1 |
| 885 | 001256 | 001602 | DF1 |
| 886 | | | |
| 887 | | | ;WDE WILL NOT RESET IN MAINT MODE |
| 888 | | | |
| 889 | 001260 | 016115 | EM10 |
| 890 | 001262 | 015525 | DH1 |
| 891 | 001264 | 001570 | DT1 |
| 892 | 001266 | 001602 | DF1 |
| 893 | | | |
| 894 | | | ;RECEIVER IN DID NOT SET OR RESET |
| 895 | | | |
| 896 | 001270 | 016157 | EM11 |
| 897 | 001272 | 015525 | DH1 |
| 898 | 001274 | 001570 | DT1 |
| 899 | 001276 | 001602 | DF1 |
| 900 | | | |
| 901 | | | ;CANNOT CLEAR BOOT ENABLE |
| 902 | | | |
| 903 | 001300 | 016236 | EM12 |
| 904 | 001302 | 015525 | DH1 |
| 905 | 001304 | 001570 | DT1 |
| 906 | 001306 | 001602 | DF1 |
| 907 | | | |
| 908 | | | ;CANNOT SET BOOT ENABLE |
| 909 | | | |
| 910 | 001310 | 016271 | EM13 |
| 911 | 001312 | 015525 | DH1 |
| 912 | 001314 | 001570 | DT1 |
| 913 | 001316 | 001602 | DF1 |
| 914 | | | |
| 915 | | | ;CANNOT SET A G-LITE OR AMLITE |
| 916 | | | |
| 917 | 001320 | 016322 | EM14 |
| 918 | 001322 | 015525 | DH1 |
| 919 | 001324 | 001570 | DT1 |
| 920 | 001326 | 001602 | DF1 |
| 921 | | | |
| 922 | | | ;CANNOT RESET A G-LITE OR AMLITE |
| 923 | | | |
| 924 | 001330 | 016370 | EM15 |
| 925 | 001332 | 015525 | DH1 |
| 926 | 001334 | 001570 | DT1 |
| 927 | 001336 | 001602 | DF1 |
| 928 | | | |
| 929 | | | ;T-3 DID NOT SET OR RESET |
| 930 | | | |
| 931 | 001340 | 016440 | EM16 |
| 932 | 001342 | 015525 | DH1 |

| | | | |
|-----|--------|--------|--|
| 933 | 001344 | 001570 | DF1 |
| 934 | 001346 | 001602 | DF1 |
| 935 | | | |
| 936 | | | ;TIME 3 NOT 1 SECOND |
| 937 | | | |
| 938 | 001350 | 016517 | EM17 |
| 939 | 001352 | 016600 | DH2 |
| 940 | 001354 | 001570 | DT1 |
| 941 | 001356 | 001606 | DF2 |
| 942 | | | |
| 943 | | | ;STATIC PANEL SWITCH ERROR |
| 944 | | | |
| 945 | 001360 | 016655 | EM20 |
| 946 | 001362 | 015525 | DH1 |
| 947 | 001364 | 001570 | DT1 |
| 948 | 001366 | 001602 | DF1 |
| 949 | | | |
| 950 | | | ;SET/RESET FAILURE ON DROUT BITS |
| 951 | | | |
| 952 | 001370 | 016730 | EM21 |
| 953 | 001372 | 015525 | DH1 |
| 954 | 001374 | 001570 | DT1 |
| 955 | 001376 | 001602 | DF1 |
| 956 | | | |
| 957 | | | ;SET/RESET FAILURE WITH FRO FLOP. |
| 958 | | | |
| 959 | 001400 | 016772 | EM22 |
| 960 | 001402 | 015525 | DH1 |
| 961 | 001404 | 001570 | DT1 |
| 962 | 001406 | 001602 | DF1 |
| 963 | | | |
| 964 | | | ;SET/RESET FAILURE WITH MINUTE FLOP(S) |
| 965 | | | |
| 966 | 001410 | 017051 | EM23 |
| 967 | 001412 | 015525 | DH1 |
| 968 | 001414 | 001570 | DT1 |
| 969 | 001416 | 001602 | DF1 |
| 970 | | | |
| 971 | | | ;15 SECOND FLOP DID NOT SET ON EXACTLY 15 PULSES |
| 972 | | | |
| 973 | 001420 | 017121 | EM24 |
| 974 | 001422 | 015525 | DH1 |
| 975 | 001424 | 001570 | DT1 |
| 976 | 001426 | 001602 | DF1 |
| 977 | | | |
| 978 | | | ;15 SECOND FLOP DID NOT RESET ON A COUNT OF 30 |
| 979 | | | |
| 980 | 001430 | 017214 | EM25 |
| 981 | 001432 | 015525 | DH1 |
| 982 | 001434 | 001570 | DT1 |
| 983 | 001436 | 001602 | DF1 |
| 984 | | | |
| 985 | | | ;30 SECOND FLOP DID NOT SET ON EXACTLY 30 PULSES |
| 986 | | | |
| 987 | 001440 | 017311 | EM26 |
| 988 | 001442 | 015525 | DH1 |

| | | | |
|------|--------|--------|---|
| 989 | 001444 | 001570 | DF1 |
| 990 | 001446 | 001602 | DF1 |
| 991 | | | |
| 992 | | | ;30 SECOND FLOP DID NOT RESET ON A COUNT OF 45/⟨15⟩'12⟩ |
| 993 | | | |
| 994 | 001450 | 017404 | EM27 |
| 995 | 001452 | 015525 | DH1 |
| 996 | 001454 | 001570 | DT1 |
| 997 | 001456 | 001602 | DF1 |
| 998 | | | |
| 999 | | | ;15 OR 30 SECOND FLOP WAS NOT CLEARED BY CLEAR 1 |
| 1000 | | | |
| 1001 | 001460 | 017501 | EM30 |
| 1002 | 001462 | 015525 | DH1 |
| 1003 | 001464 | 001570 | DT1 |
| 1004 | 001466 | 001602 | DF1 |
| 1005 | | | |
| 1006 | | | ;TIME 3 WILL NOT TOGGLE 15 SECOND COUNTER |
| 1007 | | | |
| 1008 | 001470 | 017562 | EM31 |
| 1009 | 001472 | 015525 | DH1 |
| 1010 | 001474 | 001570 | DT1 |
| 1011 | 001476 | 001602 | DF1 |
| 1012 | | | |
| 1013 | | | ;CANNOT SET BOOT TO A ONE |
| 1014 | | | |
| 1015 | 001500 | 017651 | EM32 |
| 1016 | 001502 | 015525 | DH1 |
| 1017 | 001504 | 001570 | DT1 |
| 1018 | 001506 | 001602 | DF1 |
| 1019 | | | |
| 1020 | | | ;CANNOT SET MINUTE FLOP WITH SERIAL INPUT |
| 1021 | | | |
| 1022 | 001510 | 017711 | EM33 |
| 1023 | 001512 | 015525 | DH1 |
| 1024 | 001514 | 001570 | DT1 |
| 1025 | 001516 | 001602 | DF1 |
| 1026 | | | |
| 1027 | | | ;EXTRA T1 PULSES DIT NOT CAUSE ALARM |
| 1028 | | | |
| 1029 | 001520 | 017770 | EM34 |
| 1030 | 001522 | 015525 | DH1 |
| 1031 | 001524 | 001570 | DT1 |
| 1032 | 001526 | 001602 | DF1 |
| 1033 | | | |
| 1034 | | | ;INITIALIZE DID NOT CLEAR FRO FLOP |
| 1035 | | | |
| 1036 | 001530 | 021017 | EM35 |
| 1037 | 001532 | 015525 | DH1 |
| 1038 | 001534 | 001570 | DT1 |
| 1039 | 001536 | 001602 | DF1 |
| 1040 | | | |
| 1041 | | | ;BOOT DID NOT CLEAR WITHIN 2 SEC. |
| 1042 | | | |
| 1043 | 001540 | 021063 | EM36 |
| 1044 | 001542 | 015525 | DH1 |

```

1045 001544 001570 DF1
1046 001546 001602 DF1
1047
1048 ;T1 DID NOT INTERRUPT WITH IN 48M.S.
1049
1050 001550 021137 EM37
1051 001552 015525 DH1
1052 001554 001570 DT1
1053 001556 001602 DF1
1054
1055 ;BOOT ENABLE DID NOT SET TIME 3 ENABLE.
1056
1057 001560 021211 EM40
1058 001562 015525 DH1
1059 001564 001570 DT1
1060 001566 001602 DF1
1061
1062 001570 001116 001120 001124 DT1: .WORD $ERRPC,$GDADR,$GDDAT,$BDDAT,0
1063 001576 001126 000000
1064
1065 001602 000 000 000 DF1: .BYTE 0,0,0,0
1066 001605 000
1067 001606 000 001 001 DF2: .BYTE 0,1,1,1
1068 001611 001
1069
1070 ;NW11 REGISTER 0 DEFINITIONS (17XXX0)
1071
1072 000001 TGO =BIT00
1073 000002 WOE =BIT01
1074 000004 MAINT =BIT02
1075 000010 BOOT =BIT03
1076 000020 BTENA =BIT04
1077 000040 TM3 =BIT05
1078 000100 INTENA =BIT06
1079 000200 ALARM =BIT07
1080 000400 TM1 =BIT08
1081 001000 TM2 =BIT09
1082 002000 TM3LTE =BIT10
1083 004000 PWR0UT =BIT11
1084 010000 GNLITE =BIT12
1085 020000 AMLITE =BIT13
1086 040000 RDLITE =BIT14
1087 100000 RECVIN =BIT15
1088
1089 ;NW11 REGISTER 04 DEFINITIONS (17XXX4)
1090
1091 000001 PNLCK =BIT00
1092 000004 PSW1 =BIT02
1093 000010 PSW2 =BIT03
1094 000020 PSW3 =BIT04
1095 000040 PSW4 =BIT05
1096 000100 PSW5 =BIT06
1097 000200 PSW6 =BIT07
1098 001000 TIINT =BIT09
1099 002000 PLITE1 =BIT10
1100 004000 PLITE2 =BIT11
  
```

1101 010000
1102 020000
1103 040000
1104 100000

PLITE3 =BIT12
PLITE4 =BIT13
PLITE5 =BIT14
PLITE6 =BIT15

;NW11 REGISTER 06 DEFINITIONS (17XXX6)

1109 000001
1110 000002
1111 000004
1112 000010
1113 000020
1114 000040
1115 000100
1116 000200
1117 000400
1118 001000
1119 002000
1120 004000
1121 010000
1122 020000
1123 040000
1124 100000

FROFLP =BIT00
SEC15 =BIT01
SEC30 =BIT02
MIN1 =BIT03
MIN2 =BIT04
MIN4 =BIT05
MIN8 =BIT06
SETSEC =BIT07
DR0UT1 =BIT08
DR0UT2 =BIT09
DR0UT3 =BIT10
DR0UT4 =BIT11
DR0UT5 =BIT12
DR0UT6 =BIT13
DR0UT7 =BIT14
DR0UT8 =BIT15

1126 001612 000000
1127 001614 000000
1128 001616 000000
1129 001620 000000
1130 001622 000000
1131 001624 000000
1132 001626 000000
1133 001630 000000
1134 001632 000000
1135 001634 000000
1136 001636 000000
1137 001640 000000
1138 001642 000000
1139 001644 000000
1140 001646 001656
1141 001650 000000
1142 001652 001664
1143 001654 001704
1144 001656 000004 000100 000000
1145 001664 001000 002000 004000
1146 001672 010000 020000 040000
1147 001700 100000 000000
1148 001704 000010 000020 000040
1149 001712 000100 000400 001000
1150 001720 002000 004000 010000
1151 001726 020000 040000 100000
1152 001734 000000

NCSR: .WORD ;NW11 ADDRESS CONTROL AND TIME
NVECT: .WORD ;NW11 ADDRESS 17XXX2 - VECTOR
NPNL: .WORD ;NW11 ADDRESS 17XXX4 - PANEL LITE/PANEL SWITCHES
NTIM: .WORD ;NW11 ADDRESS 17XXX6 - OUTPUT DRIVERS AND TIME
MINJMP: .WORD ;STORE MINUTE JUMPERS
BTCND: .WORD 0 ;RE-BOOT CONFIG. STORED HERE.
NOBOOT: .WORD 0 ;NO RE-BOOT CONFIG. STORED HERE.
TEMPA: .WORD ;SCRATCH LOCATION
LCLOK: .WORD ;KW11-L ADDRESS
NWVECT: .WORD ;NW11 VECTOR
NWTINT: .WORD ;NW11 T1 INT. VECTOR
ZERO: .WORD 0 ;ZERO LOCATION
NOTYP: .WORD 0 ;START/RESTART FLAG LOCATION
CLKCT: .WORD 0 ;REAL TIME CLOCK COUNTER
WREG: .WORD WBIT0
.WORD 0
.WORD WBIT4
.WORD WBIT6
WBIT0: .WORD BIT02,BIT06,0
WBIT4: .WORD BIT09,BIT10,BIT11,BIT12,BIT13,BIT14,BIT15,0
WBIT6: .WORD BIT03,BIT04,BIT05,BIT06,BIT8,BIT9,BIT10,BIT11,BIT12,BIT13,BIT14,BIT15,0

;*****
;PRINT OUT SETUP MESSAGE, ASK FOR AND STORE THE

1153
1154
1155
1156

M02

```

1157                                   ;NW11 ADDRESS.
1158
1159                                   ;*****
1160
1161   001736   012737   CJ0001   001642   LSTART: MOV    #1,NOTYP           ;DO NOT TYPE HEADER INFO
1162   001744   000402                   BR       SSTART
1163   001746   005037   J01642           NSTART: CLR    NOTYP           ;FULL TEST
1164   001752                   SSTART:
1165                   .SBTTL   INITIALIZE THE COMMON TAGS
1166                   ;CLEAR THE COMMON TAGS ($CMTAG) AREA
1167   001752   012706   001100           MOV    #SCMTAG,R6           ;FIRST LOCATION TO BE CLEARED
1168   001756   005026                   CLR    (R6)+               ;CLEAR MEMORY LOCATION
1169   001760   022706   001140           CMP    #SWR,R6   ;;DONE?
1170   001764   001374                   BNE                       ;LOOP BACK IF NO
1171   001766   012706   001100           MOV    #STACK,SP           ;SETUP THE STACK POINTER
1172                   ;INITIALIZE A FEW VECTORS
1173   001772   012737   011742   000020   MOV    #$SCOPE,#IOTVEC   ;;IOT VECTOR FOR SCOPE ROUTINE
1174   002000   012737   000340   000022   MOV    #340,#IOTVEC+2   ;LEVEL 7
1175   002006   012737   012212   000030   MOV    #$ERROR,#EMTVEC   ;EMT VECTOR FOR ERROR ROUTINE
1176   002014   012737   000340   000032   MOV    #340,#EMTVEC+2   ;LEVEL 7
1177   002022   012737   014312   000034   MOV    #$TRAP,#TRAPVEC   ;TRAP VECTOR FOR TRAP CALLS
1178   002030   012737   000340   000036   MOV    #340,#TRAPVEC+2;LEVEL 7
1179   002036   012737   014134   000024   MOV    #$PWRDN,#PWRVEC   ;POWER FAILURE VECTOR
1180   002044   012737   000340   000026   MOV    #340,#PWRVEC+2   ;LEVEL 7
1181   002052   013737   011320   011312   MOV    $ENDCT,$EOPCT    ;SETUP END-OF-PROGRAM COUNTER
1182   002060   005037   001160           CLR    $TIMES            ;INITIALIZE NUMBER OF ITERATIONS
1183   002064   005037   001162           CLR    $ESCAPE           ;CLEAR THE ESCAPE ON ERROR ADDRESS
1184   002070   112737   000001   001115   MOVB   #1,$ERMAX        ;ALLOW ONE ERROR PER TEST
1185   002076   012737   002076   001106   MOV    #,$SLPADR        ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1186   002104   012737   002104   001110   MOV    #,$SLPERR        ;SETUP THE ERROR LOOP ADDRESS
1187                   ;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1188                   ;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1189   002112   013746   000004           MOV    #ERRVEC,-(SP)    ;SAVE ERROR VECTOR
1190   002116   012737   002152   000004   MOV    #64$,#ERRVEC    ;SET UP ERROR VECTOR
1191   002124   012737   177570   001140   MOV    #DSWR,SWR        ;SETUP FOR A HARDWARE SWICH REGISTER
1192   002132   012737   177570   001142   MOV    #DDISP,DISPLAY   ;AND A HARDWARE DISPLAY REGISTER
1193   002140   022777   177777   176772   CMP    #-1,$SWR        ;TRY TO REFERENCE HARDWARE SWR
1194   002146   001012                   BNE    66$               ;BRANCH IF NO TIMEOUT TRAP OCCURRED
1195                   ;AND THE HARDWARE SWR IS NOT = -1
1196   002150   000403                   BR     65$               ;BRANCH IF NO TIMEOUT
1197   002152   012716   002160           64$: MOV    #65$,(SP)        ;SET UP FOR TRAP RETURN
1198   002156   000002                   RTI
1199   002160   012737   000176   001140   65$: MOV    #SWREG,SWR    ;POINT TO SOFTWARE SWR
1200   002166   012737   000174   001142   MOV    #DISPREG,DISPLAY
1201   002174   012637   070004           66$: MOV    (SP)+,#ERRVEC   ;RESTORE ERROR VECTOR
1202
1203   002200   005737   001642           TST    NOTYP            ;BYPASS HEADER INFO?
1204   002204   001402                   BEQ    70$               ;NO
1205   002206   000137   002670           JMP    TSTLOP           ;YES,NO HEADER,CLOCK OR VECTOR CHECKS
1206
1207   002212   104400   001165           70$: TYPE   ,$SCRLF        ;TYPE A CARRIAGE RETURN.
1208   002216   104400   014354           GTADD: TYPE   ,$DSETUP   ;ISSUE INITAL MESSAGE
1209
1210   002222   104407                   TPSTRT: RDOCT
1211
1212   002224   012601                   MOV    (SP)+,R1        ;POP STACK INTO R1

```

```

1213 002226 022701 160000          CMP      #160000,R1      ;ADDRESS LEGAL?
1214 002232 101371          BHI      GTADD          ;NO-REQUEST ANOTHER.
1215 002234 010137 001612          MOV      R1,NCSR
1216 002240 062701 000002          ADD      #2,R1
1217 002244 010137 001614          MOV      R1,NCSR+2
1218 002250 062701 000002          ADD      #2,R1
1219 002254 010137 001616          MOV      R1,NCSR+4      ;LOAD THE NW11 ADDRESSES
1220 002260 062701 000002          ADD      #2,R1
1221 002264 010137 001620          MOV      R1,NCSR+6
1222
1223      ;*****
1224
1225      ;ROUTINE TO INSURE OPERATION OF CLOCK
1226      ;*****
1227
1228 002270 012737 177546 001632 PASSLP: MOV      #177546,LCLOCK ;SET UP CLOCK ADDRESS
1229 002276 012737 002320 000004      MOV      #CKOUT,#4      ;SET UP CLOCK TIMEOUT ADDRESS
1230 002304 012737 000340 000006      MOV      #340,#6
1231 002312 005777 177314          TST      @LCLOCK        ;TEST THE CLOCK
1232 002316 000405          BR       TIMCK          ;MUST BE PRESENT GO TIME IT.
1233
1234 002320 104400 014703          CKOUT:  TYPE ,DNOCLK    ;COULD NOT ADDRESS CLOCK ISSUE
1235 002324 000000          HALT                                ;TIMEOUT MESSAGE AND HALT
1236 002326 000137 001746          JMP      NSTART
1237
1238 002332 012737 011610 001630 TIMCK:  MOV      #5000.,TEMPA ;SET UP TO DELAY AT LEAST 18M.S.
1239 002340 005077 177266          CLR      @LCLOCK        ;CLEAR THE CLOCK
1240 002344 005777 177262          IS:    TST      @LCLOCK ;CLECK SET?
1241 002350 001010          BNE      SETVEC         ;YES EXIT
1242 002352 005337 001630          DEC      TEMPA          ;NO RUN TIME OUT DELAY
1243 002356 001372          BNE      IS             ;AND TRY AGAIN
1244 002360 104400 015035          TYPE    ,DNORN         ;TIMED OUT ISSUE CLOCK NO OR
1245 002364 000000          HALT                                ;SLOW RUN MESSAGE AND HALT
1246 002366 000137 001746          JMP      NSTART
1247 002372 012737 011410 000100 SETVEC: MOV      #TCLOCK,#100 ;SET UP CLOCK VETOR AND PSW
1248 002400 012737 000340 000102      MOV      #340,#102
1249
1250      ;*****
1251
1252      ;BEGIN TEST BY READING AND DISPLAYING VECTOR INSURING
1253      ;THE NW11 CAN BE ADDRESSED AND THAT DATA OUT IS NOT
1254      ;HIGH IN THE NW11. IF DATA OUT IS HIGH, TEST WILL HALT AT
1255      ;THE VECTOR ADDRESS.
1256
1257      ;*****
1258
1259 002406 ADDVEC:
1260      ;*****
1261 002406 000004      TST1:  SCOPE
1262 002410 012737 000001 001160      MOV      #1,$TIMES     ;;DO 1 ITERATION
1263 002416 012737 002424 001106      MOV      #TSTADD,$LPADR ;;SET SCOPE LOOP ADDRESS
1264
1265 002424 012737 002440 000004 TSTADD: MOV      #NWOUT,#4   ;SET UP ADDRESS TIMEOUT
1266 002432 005777 177154          TST      @NCSR          ;CAN WE ADDRESS THE NW11?
1267 002436 000404          BR       CKVECT        ;YES GO READ VECTOR
1268

```

```

1269 002440 104400 015151          NJOUT: TYPE      ,DNONW          ;NO ISSUE TRAP MESSAGE AND
1270 002444 000000                HALT                    ;HALT
1271 002446 000665                BR          TPSTRT
1272
1273 002450 017737 177140 001634 CKVECT: MOV      @NVECT,NWVECT ;READ THE VECTOR
1274 002456 104400 015226                TYPE      ,DVECT          ;AND DISPLAY IT
1275 002462 013746 001634                MOV      NWVECT,-(SP)      ;SAVE NWVECT FOR TYPEOUT
1276 002466 104402                TYPOS                    ;GO TYPE--OCTAL ASCII
1277 002470          006                .BYTE      6              ;TYPE 6 DIGITS
1278 002471          000                .BYTE      0              ;SUPPRESS LEADING ZEROS
1279 002472 104400 001165                TYPE      ,SCLRF          ;AND A CARRIAGE RETURN
1280 002476 012737 000006 000004                MOV      #6,@#4
1281 002504 005037 000006                CLR      @#6              ;RESTORE TIME OUT HALT
1282 002510 013737 001634 001636                MOV      NWVECT,NWINT     ;BUILD THE TI INTERRUPT VECTOR
1283 002516 062737 000004 001636                ADD      #4,NWINT
1284 002524 052777 000004 177060                BIS      @MAINT,@NCSR     ;ENTER MAINT MODE TO AVOID BOOT.
1285
1286 ;*****
1287
1288 ;ROUTINE TO READ THE STATE OF THE WATCH DOG
1289 ;REGISTERS AFTER A SYSTEM RESET. ALL REGISTERS
1290 ;SHOULD CONTAIN ZEROS EXCEPT REG.0, BIT 3 WHICH
1291 ;IS CONSTANTLY TOGGLING; REG.0,BIT 10 WHICH WILL
1292 ;BE A 1; AND REG 2 WHICH IS VECTOR.
1293
1294 ;*****
1295
1296 ;*****
1297
1298 002532 000004                TST2: SCOPE
1299 002534 012737 000001 001160                MOV      #1,STIMES       ;DO 1 ITERATION
1300 002542 012737 002550 001106                MOV      @RREG,@LPADR    ;SET SCOPE LOOP ADDRESS
1301 002550 104400 015250                RDRREG: TYPE      ,DSHTDN ;TYPE SHUT DOWN MESSAGE
1302 002554 000001                WAIT                    ;AND WAIT FOR IT TO OCCUR
1303 002556 104400 001165                TYPE      ,SCLRF
1304 002562 013737 001634 002662                MOV      NWVECT,RDRD+2
1305
1306 002570 005002                REGLP: CLR      R2          ;CLEAR OFFSET POINTER
1307 002572 017203 001612                MOV      @NCSR(R2),R3    ;LOAD REG. READ INTO TEMP. STORAGE.
1308 002576 020362 002660                CMP      R3,RDRD(R2)     ;COMPARE READ TO GOOD
1309 002602 001421                BEQ      RDNXT           ;OK - CHECK ITERATION
1310 002604 005702                TST      R2              ;FIRST ITERATION(REG 0)?
1311 002606 001005                BNE      IS              ;NO-ERROR OCCURRED.
1312 002610 042703 000040                BIC      @TM3,R3         ;YES-ERROR MIGHT BE TIME 3 CLOCKING.
1313 002614 020362 002660                CMP      R3,RDRD(R2)     ;GOOD COMPARE NOW?
1314 002620 001412                BEQ      RDNXT           ;YES- SETUPFOR REG 1.
1315 002622 016237 001612 001120 IS:      MOV      NCSR(R2),@GDADR ;SET UP AND ISSUE ERROR1
1316 002630 016237 002660 001124                MOV      RDRD(R2),@GDADR
1317 002636 017237 001612 001126                MOV      @NCSR(R2),@BDADR
1318 002644 104001                ERROR 1
1319 002646 005722                RDNXT: TST      (R2)+     ;INCREMENT OFFSET
1320 002650 022702 000004                CMP      #4,R2           ;OFFSET =4?
1321 002654 001346                BNE      REGLP           ;NO-LOOP
1322 002656 000404                BR       TSTLOP         ;YES-EXIT
1323
1324 002660 002000                RDRD:  .WORD      2000   ;REG 0

```

```

1325 002662 000000          .WORD 0          ;REG2 VECTOR
1326 002664 000000          .WORD 0          ;REG3
1327 002666 000000          .WORD 0          ;REG4
1328
1329 002670 052777 000004 176714 TSTLOP: BIS      #MAINT, @NCSR ;ENTER MAINT MODE TO AVOID BOOT.
1330
1331 ;*****
1332 ;ROUTINE TO SET ALL WRITE BITS INDIVIDUALLY IN
1333 ;EACH REGISTER AND INSURE THEY ARE SET BY READING
1334 ;THE REGISTER.
1335 ;*****
1336 ;*****
1337 ;*****
1338 ;*****
1339 ;*****
1340 002676 000004          TST3: SCOPE
1341 002700 012737 000005 001160      MOV      #5, $TIMES ;DO 5 ITERATIONS
1342 002706 012737 002714 001106      MOV      @WTREG, $LPADR ;SET SCOPE LOOP ADDRESS
1343
1344 002714 005001          WTREG: CLR      R1      ;CLEAR REGISTER OFFSET
1345 002716 005002          NWREG: CLR     R2      ;CLEAR BIT OFFSET
1346
1347 002720 016102 001646          NMBIT: MOV      WREG(R1), R2 ;LOAD THE REG W-BIT ADDRESS
1348 002724 051271 001612          BIS      (R2), @NCSR(R1) ;SET THE BIT
1349 002730 031271 001612          BIT      (R2), @NCSR(R1) ;IS IT SET
1350 002734 001013          BNE      IS          ;YES UPDATE BIT COUNT
1351 002736 004037 011444          JSR      RD, CLREG ;NO-CLEAR ERROR REGISTERS
1352 002742 016137 001612 001120      MOV      NCSR(R1), $GDADR ;AND LOAD THEM WITH MESSAGE
1353 002750 051237 001124          BIS      (R2), $GDADR ;DATA.
1354 002754 017137 001612 001126      MOV      @NCSR(R1), $BODAT
1355 002762 104001          ERROR 1 ;ISSUE ERROR
1356
1357 002764 041271 001612          IS:     BIC      (R2), @NCSR(R1) ;CLEAR THE BIT JUST SET.
1358 002770 005722          TST      (R2)+ ;BUMP THE BIT POINTER
1359 002772 005712          TST      (R2) ;NEXT BIT A ZERO?
1360 002774 001353          BNE      NMBIT ;NO-KEEP WORKING THIS REGISTER.
1361 002776 005721          TST      (R1)+ ;YES-UPDATE REGISTER OFFSET
1362 003000 020127 000002          2$:    CMP      R1, #2 ;AT VECTOR?
1363 003004 001004          BNE      3$ ;NO-CONTINUE WITH NEXT REGISTER.
1364 003006 052777 000004 176576      BIS      #MAINT, @NCSR ;YES-SET THE MAINT. MODE AND BYPASS REG2
1365 003014 000770          BR      2$
1366 003016 020127 000010          3$:    CMP      R1, #10 ;ALL DONE?
1367 003022 001335          BNE      NWREG ;NO CYCLE NEXT REGISTER.
1368
1369 ;*****
1370 ;ROUTINE TO VERIFY THE OPERATION OF T2 AND T2.
1371 ;TIMES ARE CHECKED TO SET AND RESET AND THAT
1372 ;T1 < T2
1373 ;*****
1374 ;*****
1375 ;*****
1376 003024 000004          TST4: SCOPE
1377 003026 012737 000005 001160      MOV      #5, $TIMES ;DO 5 ITERATIONS
1378 003034 012737 003042 001106      MOV      #T12TST, $LPADR ;SET SCOPE LOOP ADDRESS
1379 003042 013737 001612 001120      T12TST: MOV     NCSR, $GDADR ;PRELOAD REGISTER IN ERROR MESSAGE
1380 003050 004037 011720          JSR      RD, GOCLK ;START THE CLOCK.

```

```

1381 003054 052777 000004 176530      BIS      #MAINT, @NCSR      ;ENTER MAINTENANCE MODE
1382 003062 052777 000001 176522      BIS      #TGO, @NCSR      ;START T1 AND T2
1383 003070 000240      NOP
1384 003072 000240      NOP      ;GIVE THEM TIME TO SET
1385 003074 032777 000400 176510      BIT      #TM1, @NCSR      ;T1 SET?
1386 003102 001404      BEQ      1$      ;NO ERROR
1387 003104 032777 001000 176500      BIT      #TM2, @NCSR      ;T2 SET?
1388 003112 001012      BNE      2$      ;T1 AND T2 SET.
1389 003114 017737 176472 001126 1$:      MOV      @NCSR, $BDDAT
1390 003122 013737 001612 001120      MOV      NCSR, $GDADR      ;ISSUE T1 OR T2 DID NOT
1391 003130 012737 043634 001124      MOV      #43634, $GDDAT      ;SET MESSAGE
1392 003136 104002      ERROR      2
1393
1394 003140 123727 001645 000003 2$:      CMPB     CLKCT+1, #3      ;INSURE T1 AND T2 BOTH
1395 003145 103413      BLO      3$      ;RESET WITHIN 3 SECONDS.
1396 003150 013737 001612 001120      MOV      NCSR, $GDADR
1397 003156 017737 176430 001126      MOV      @NCSR, $BDDAT
1398 003164 012737 042224 001124      MOV      #42224, $GDDAT
1399 003172 104003      ERROR      3
1400 003174 000404      BR       T12TB
1401 003176 032777 001400 176406 3$:      BIT      #TM1+TM2, @NCSR
1402 003204 001355      BNE      2$
1403
1404 003206 052777 000001 176376 T12TB:  BIS      #TGO, @NCSR      ;RESTART T1 AND T2
1405 003214 032777 000400 176370 1$:      BIT      #TM1, @NCSR
1406 003222 001374      BNE      1$      ;WAIT FOR T1 TO RESET
1407 003224 032777 001000 176360      BIT      #TM2, @NCSR
1408 003232 001012      BNE      2$      ;T2 SHOULD STILL BE SET
1409 003234 013737 001612 001120      MOV      NCSR, $GDADR
1410 003242 017737 176344 001126      MOV      @NCSR, $BDDAT      ;IT WASN'T ISSUE T1
1411 003250 012737 043234 001124      MOV      #43234, $GDDAT      ;T2 TIMING ERROR
1412 003256 104004      ERROR      4
1413 003260 005037 001644 2$:      CLR      CLKCT
1414 003264 123727 001645 000003 3$:      CMPB     CLKCT+1, #3      ;ALLOW TIME FOR T2 TO
1415 003272 103774      BLO      3$      ;RESET BEFORE EXITING.
1416
1417      ;;*****
1418      ;ROUTINE TO INSURE THAT ALARM SETS IF T1,T2
1419      ;SEQUENCE IS NOT SUPPORTED. ALARM IS THEN RESET
1420      ;BY SUPPORTING THE T1,T2 SEQUENCE (SETTING CLR1).
1421      ;;*****
1422
1423      ;;*****
1424 003274 000004      T$T5:    SCOPE
1425 003276 012737 000005 001160      MOV      #5, $TIMES      ;;DO 5 ITERATIONS
1426 003304 012737 003312 001106      MOV      @T$TALR, $LPAOR ;;T SCOPE LOOP ADDRESS
1427
1428 003312 052777 000004 176272 T$TALR:  BIS      #MAINT, @NCSR      ;SET MAINTENANCE MODE TO AVOID BOOTS.
1429 003320 004037 011720      JSR      RD, GOCLK      ;START THE CLOCK.
1430 003324 052777 000001 176260      BIS      #TGO, @NCSR      ;START T1 AND T1
1431 003332 032777 001000 176252 1$:      BIT      #TM2, @NCSR
1432 003340 001374      BNE      1$      ;LOOP UNTIL T2 TIMES OUT
1433 003342 032777 000200 176242      BIT      #ALARM, @NCSR      ;ALARM SET?
1434 003350 001012      BNE      CLRALR      ;YES GO TO NEXT SUBTEST
1435 003352 013737 001612 001120      MOV      NCSR, $GDADR
1436 003360 017737 176226 001126      MOV      @NCSR, $BDDAT      ;NO ISSUE NO ALARM OR

```

E03

MAINDEC-11-DZMWA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 31
 DZMWA.SRC INITIALIZE THE COMMON TAGS

```

1437 003366 012737 042244 001124      MOV      #42244,$GDDAT ;FROFLP 'ERROR.
1438 003374 104005                      ERROR      5
1439
1440 003376 012701 000005      CLRALR: MOV      #5,R1
1441 003402 052777 000001 176202 15:      BIS      #TGO,ANCSR ;START AND SUPPORT T1,T2
1442 003410 032777 000400 176174 25:      BIT      #TM1,ANCSR ;SEQUENCES FOR A COUNT
1443 003416 001374                      BNE      25 ;OF FIVE
1444 003420 005301                      DEC      R1
1445 003422 001367                      BNE      15
1446 003424 032777 000200 176160      BIT      #ALARM,ANCSR ;IS 'ALARM' NOW CLEARED?
1447 003432 001412                      BEQ      ALRXT ;YES-EXIT
1448 003434 013737 001612 001120      MOV      NCSR,$GDAOR
1449 003442 017737 176144 001126      MOV      ANCSR,$BDDAT ;LOAD AND ISSUE 'ALARM OR
1450 003450 012737 031054 001124      MOV      #31054,$GDDAT ;GLITE DID NOT CLEAR' ERROR
1451 003456 104006                      ERROR      6
1452 003460 032777 001000 176124 ALRXT: BIT      #TM2,ANCSR ;WAIT FOR T-2 TO TIME OUT
1453 003466 001374                      BNE      ALRXT ;BEFORE EXITING
1454
1455 ;*****
1456 ;ROUTINE TO INSURE WOE CAN BE SET BY
1457 ;BIT 1 WHEN UNIT IS IN MAINTENANCE MODE.
1458 ;SECTION B INSURES THIS FLAG CANNOT BE SET
1459 ;IF NOT IN MAINT.
1460 ;*****
1461
1462 ;*****
1463 003470 000004      TSTB:  SCOPE
1464 003472 012737 000005 001160      MOV      #5,STIMES ;DO 5 ITERATIONS
1465 003500 012737 003506 001106      MOV      #SET0,$LPADR ;SET SCOPE LOOP ADDRESS
1466
1467 003506 052777 000004 176076 SET0:  BIS      #MAINT,ANCSR ;SET MAINT MODE
1468 003514 052777 000002 176070      BIS      #WDE,ANCSR ;SET WDE
1469 003522 032777 000002 176062      BIT      #WDE,ANCSR ;DID WATCH DOG ENABLE SET?
1470 003530 001007                      BNE      RSETA ;YES TO NEXT SUBTEST
1471 003532 017737 176054 001126      MOV      ANCSR,$BDDAT
1472 003540 012737 042236 001124      MOV      #42236,$GDDAT ;NO ISSUE WATCH DOG ENABLE
1473 003546 104007                      ERROR      7 ;OR T3 ENABLE DID NOT SET.
1474 003550 004037 011534      RSETA: JSR      RO,FORCT ;CLEAR THE ALARM
1475 003554 032777 000010 176030      BIT      #BOOT,ANCSR ;IF T1 IS SET SHORT, A BOOT
1476 003562 001374                      BNE      -6 ;COULD OCCUR-TRAP IT.
1477 003564 042777 000004 176020      BIC      #MAINT,ANCSR ;RESET MAINT AND INSURE
1478 003572 052777 000002 176012      BIS      #WDE,ANCSR ;WDE CANNOT BE SET.
1479 003600 032777 000002 176004      BIT      #WDE,ANCSR ;IS WDE RESET?
1480 003606 001412                      BEQ      EXITA ;YES-EXIT.
1481 003610 052777 000004 175774      BIS      #MAINT,ANCSR ;SET MAINT. TO PROHIBIT A BOOT.
1482 003616 017737 175770 001126      MOV      ANCSR,$BDDAT ;ONE OR BOTH DID NOT RESET
1483 003624 012737 040200 001124      MOV      #40200,$GDDAT ;ISSUE ERROR
1484 003632 104010                      ERROR      10
1485 003634 052777 000004 175750 EXITA: BIS      #MAINT,ANCSR ;INSURE NO BOOTS
1486 003642 032777 001000 175742      BIT      #TM2,ANCSR ;LOOP UNTIL TIME 2 EXPIRES.
1487 003650 001374                      BNE      -6
1488
1489 ;*****
1490 ;ROUTINE TO CHECK RECV. IN VIA SETTING DROUT 1.
1491 ;MAINT. SWITCH MUST BE IN ON POSITION FOR THIS TEST
1492 ;TO WORK.

```

```

1493 ; ;*****
1494
1495 ; ;*****
1496 003652 000004
1497 003654 012737 000005 001160
1498 003662 012737 003670 001106
1499 003670 052777 000004 175714
1500 003676 004037 011720
1501 003702 052777 000400 175710
1502 003710 032777 100000 175674
1503 003716 001005
1504 003720 123727 001644 000002
1505 003726 001411
1506 003730 000767
1507
1508 003732 032777 100000 175652
1509 003740 001413
1510 003742 123727 001644 000002
1511 003750 103770
1512 003752 017737 175634 001126
1513 003760 012737 142234 001124
1514 003766 104011
1515 003770 000240
1516
1517 ; ;*****
1518 ;ROUTINE TO INSURE BOOT ENABLE CAN BE SET BY
1519 ;MAINT + 'BOO4' AND BY 'RECV. IN.
1520 ;INSURE 'CLR1' WILL RESET BOOT ENA.
1521
1522 ; ;*****
1523
1524 ; ;*****
1525 003772 000004
1526 003774 012737 000005 001160
1527 004002 012737 004010 001106
1528
1529 004010 052777 000004 175574
1530 004016 004037 011720
1531 004022 004037 011530
1532 004026 032777 000020 175556
1533 004034 001410
1534 004036 017737 175550 001126
1535 004044 012737 040204 001124
1536 004052 104012
1537 004054 000507
1538 004056 052777 000020 175526
1539 004064 032777 000020 175520
1540 004072 001414
1541 004074 032777 002000 175510
1542 004102 001017
1543 004104 017737 175502 001126
1544 004112 012737 042224 001124
1545 004120 104040
1546 004122 000407
1547 004124 017737 175462 001126
1548 004132 012737 042224 001124

```

```

; ;*****
; ;*****
TST7: SCOPE
MOV #5,STIMES ;DO 5 ITERATIONS
MOV #RECVCK,$LPADR ;SET SCOPE LOOP ADDRESS
RECVCK: BIS #MAINT,#NCSR ;SET MAINTENANCE MODE.
JSR RO,GOCCLK ;START THE CLOCK.
BIS #DROUT1,#NTIM ;SET THE 'ONE:
1$: BIT #RECVIN,#NCSR ;LOOK FOR IT LOOPED INTO RECV.IN.
BNE RECVA ;RECEIVE 'ONE' EXIT
CMPB CLKCT,#2 ;COUNTED TO 32 M.S.?
BEQ RECERR ;YES ERROR
BR 1$ ;NO KEEP LOOKING

RECVA: BIT #RECVIN,#NCSR ;NOW WAIT FOR 'ONE' AND
BEQ RECVXT ;RECV. IN TO CLEAR.
CMPB CLKCT,#2 ;COUNTED TO 32 M.S.?
BLO RECVA ;NO KEEP LOOPING
RECERR: MOV #NCSR,$BDDAT ;YES ISSUE RECEIVER IN
MOV #142234,$GDDAT ;ERROR
RECVXT: NOP

; ;*****
;ROUTINE TO INSURE BOOT ENABLE CAN BE SET BY
;MAINT + 'BOO4' AND BY 'RECV. IN.
;INSURE 'CLR1' WILL RESET BOOT ENA.

; ;*****
; ;*****
TST10: SCOPE
MOV #5,STIMES ;DO 5 ITERATIONS
MOV #TSTBEN,$LPADR ;SET SCOPE LOOP ADDRESS

TSTBEN: BIS #MAINT,#NCSR ;INSURE MAINT. MODE SET.
JSR RO,GOCCLK ;START THE CLOCK.
JSR RO,FORCTA ;GENERATE A CLEAR PULSE.
BIT #BTENA,#NCSR ;INSURE BOOT ENABLE CLEAR
BEQ 1$
MOV #NCSR,$BDDAT
MOV #40204,$GDDAT ;NOT CLEAR ISSUE ERROR.
ERROR 12
BR RSTBEN ;GO TRY ANOTHER RESET.
1$: BIS #BTENA,#NCSR ;TRY TO FORCE BOOT ENABLE.
BIT #BTENA,#NCSR ;BOOT ENABLE SET?
BEQ 10$ ;NO-ISSUE ERROR.
BIT #TM3L TE,#NCSR ;DID BOOT ENABLE SET T3 ENABLE?
BNE 2$ ;YES GO TO NEXT SET ATTEMPT
MOV #NCSR,$BDDAT ;NO-LOAD AND ISSUE CANT SET T3 ENABLE.
MOV #042224,$GDDAT
ERROR 40
BR 2$
10$: MOV #NCSR,$BDDAT ;NO ISSUE 'CAN'T SET
MOV #042224,$GDDAT ;BOOT ENABLE ERROR.

```

```

1549 004140 104013          ERROR 13
1550
1551 004142 004037 011530      2$: JSR   RD, FORCTA      ; GENERATE A CLEAR PULSE.
1552 004146 005037 001644          CLR   CLKCT          ; CLEAR CLOCK COUNT BEFORE TIMING.
1553 004152 052777 000400 175440    BIS   #DROUT1, @NTIM
1554 004160 032777 100000 175424    20$: BIT   #RECVIN, @NCSR      ; FORCE RECV. IN AND WAIT
1555 004166 001013          BNE   3$             ; FOUND IT CONTINUE.
1556 004170 123727 001645 000001    CMPB  CLKCT+1, #1     ; COUNTED TO 1 SEC.?
1557 004176 103770          BLO   20$           ; NO-KEEP LOOKING.
1558 004200 017737 175406 001126    MOV   @NCSR, $BDDAT   ; YES-LOAD AND ISSUE
1559 004206 012737 140224 001124    MOV   #140224, $GDDAT ; CANT SET REC. IN ERROR.
1560 004214 104011          ERROR 11
1561 004216 032777 000020 175366    3$: BIT   #BTENA, @NCSR   ; BOOTS ENABLE SET?
1562 004224 001413          BEQ   5$             ; NO-ISSUE ERROR AND EXIT
1563 004226 032777 000400 175364    4$: BIT   #DROUT1, @NTIM   ; YES-SET UP TO
1564 004234 001417          BEQ   RSTBEN        ; EXIT WHEN DROUT1 TIMES OUT.
1565 004236 123727 001645 000002    CMPB  CLKCT+1, #2     ; OR TYPE ERROR IF IT DOESNT.
1566 004244 103770          BLO   4$
1567 004246 104400 020037          TYPE  , DDR1
1568 004252 000434          BR    BENEXT
1569 004254 017737 175332 001126    5$: MOV   @NCSR, $BDDAT   ; NO ISSUE 'CAN'T SET BOOT
1570 004262 012737 142222 001124    MOV   #142222, $GDDAT ; ,LHABLE ERROR
1571 004270 104013          ERROR 13
1572 004272 000424          BR    BENEXT
1573
1574 004274 032777 000020 175310  RSTBEN: BIT   #BTENA, @NCSR ; WAS BOOT ENABLE EVER SET?
1575 004302 001003          BNE   2$             ; YES GO TO RESET TESTS
1576 004304 104400 015336          TYPE  , DNSEBT       ; NO TYPE COULD NOT SET
1577 004310 000415          BR    BENEXT        ; BOOT ENABLE' AND EXIT
1578
1579 004312 004037 011530      2$: JSR   RD, FORCTA      ; GENERATE A CLEAR PULSE.
1580 004316 032777 000020 175266    BIT   #BTENA, @NCSR   ; DID CLR 1 RESET BOOT ENABLE?
1581 004324 001407          BEQ   BENEXT        ; YES EXIT
1582 004326 017737 175260 001126    MOV   @NCSR, $BDDAT   ; NO ISSUE 'CAN'T RESET BOOT
1583 004334 012737 040224 001124    MOV   #040224, $GDDAT ; ENABLE 'ERROR
1584 004342 104012          ERROR 12
1585
1586 004344 000240          BENEXT: NOP
1587
1588      ; ; *****
1589      ; ROUTINE TO INSURE GREEN AND AMBER LIGHT CIRCUITS
1590      ; ARE ENABLED WHEN T1-T2 SEQUENCE IS SUPPORTED AND
1591      ; PANEL WATCH DOG ENABLE IS OFF.
1592
1593      ; ; *****
1594
1595      ; ; *****
1596      ; TST1: SCOPE
1597 004346 000004          MOV   #5, $TIMES     ; DO 5 ITERATIONS
1598 004350 012737 000005 001160    MOV   #TSTGA, $LPADR ; SET SCOPE LOOP ADDRESS
1599 004356 012737 004364 001106
1600 004364 052777 000004 175220  TSTGA: BIS   #MAINT, @NCSR ; SET MAINTENANCE MODE TO AVOID BOOTS.
1601 004372 012701 000005          MOV   #5, R1
1602 004376 052777 000001 175206    1$: BIS   #TGO, @NCSR   ; SUPPORT T1-T2 SEQUENCE
1603 004404 032777 000400 175200    BIT   #TMI, @NCSR    ; FOR A COUNT OF 5
1604 004412 001374          BNE   .-6

```

```

1605 004414 005301          DEC      R1
1606 004416 001367          BNE     15
1607 004420 032777 010000 175164  BIT     #GNLITE, @NCSR ; GREEN LITE SET?
1608 004426 001404          BEQ     25 ; NO TO ERROR
1609 004430 032777 020000 175154  BIT     #AMLITE, @NCSR ; YES IS AMBER LITE SET?
1610 004436 001007          BNE     RESGA ; YES GO TO RESET TEST
1611 004440 017737 175146 001126 25:  MOV     @NCSR, $BDDAT ; NO ISSUE CAN'T SET
1612 004446 012737 031044 001124  MOV     #031044, $GDDAT ; A GLITE OR AN AMLITE' MESSAGE.
1613 004454 104014          ERROR   14
1614
1615 004456 032777 001000 175126 RESGA: BIT     #TM2, @NCSR ; WAIT FOR TIME 2 TO TIMEOUT.
1616 004464 001374          BNE     .-6
1617 004466 000240          NOP
1618 004470 000240          NOP ; ALLOW ERROR CONDITION TO
1619 004472 032777 030000 175112  BIT     #GNLITE+AMLITE, @NCSR ; BOTH LITES RESET?
1620 004500 001407          BEQ     GAEXT ; YES EXIT
1621 004502 017737 175104 001126  MOV     @NCSR, $BDDAT ; NO ISSUE 'CAN'T REST A
1622 004510 012737 040204 001124  MOV     #40204, $GDDAT ; GLITE OR AMLITE' ERROR
1623 004516 104015          ERROR   15
1624 004520 005037 001120  GAEXT: CLR     $GDDADR ; CLEAR ITERATION COUNT
1625
1626 ;*****
1627 ;ROUTINE TO TIME T3 AND INSURE IT IS 1 SECOND
1628 ;PLUS OR MINUS 10 PER CENT.
1629 ;*****
1630
1631 ;*****
1632 004524 000004  TST12: SCOPE
1633 004526 012737 000010 001160  MOV     #10, $TIMES ; DO 10 ITERATIONS
1634 004534 012737 004542 001106  MOV     #TIME3, $LPAOR ; SET SCOPE LOOP ADDRESS
1635
1636 004542 052777 000004 175042 TIME3: BIS     #MAINT, @NCSR ; SET MAINTENANCE MODE TO AVOID BOOTS.
1637 004550 004037 011720          JSR     RD, GOCLK ; START THE CLOCK.
1638 004554 032777 000040 175030 15:  BIT     #TM3, @NCSR
1639 004562 001014          BNE     25 ; ALLOW 4 SECONDS FOR T3 TO SET
1640 004564 123727 001645 000004  CMPB   CLKCT+1, #4
1641 004572 103770          BLO     15
1642 004574 017737 175012 001126  MOV     @NCSR, $BDDAT ; T3 DID NOT SET ISSUE
1643 004602 012737 040244 001124  MOV     #40244, $GDDAT ; CAN'T SET ERROR.
1644 004610 104016          ERROR   16
1645 004612 000503          BR     TM3EXT
1646 004614 032777 000040 174770 25:  BIT     #TM3, @NCSR
1647 004622 001414          BEQ     35 ; ALLOW 4 SECONDS TO RESET T-3
1648 004624 123727 001645 000010  CMPB   CLKCT+1, #8.
1649 004632 103770          BLO     25
1650 004634 017737 174752 001126  MOV     @NCSR, $BDDAT ; T3 DID NOT RESET, ISSUE
1651 004642 012737 040204 001124  MOV     #40204, $GDDAT ; CAN'T RESET ERROR MESSAGE.
1652 004650 104016          ERROR   16
1653 004652 000463          BR     TM3EXT
1654
1655 004654 032777 000040 174730 35:  BIT     #TM3, @NCSR
1656 004662 001374          BNE     .-6 ; WAIT FOR T3 TO RESET
1657 004664 032777 000040 174720  BIT     #TM3, @NCSR
1658 004672 001774          BEQ     .-6 ; AND THEN SET
1659 004674 005037 001644          CLR     CLKCT ; CLEAR CLOCK TIMER
1660 004700 032777 000040 174704  BIT     #TM3, @NCSR

```

```

1661 004706 001374      BNE      .-6      ;AND WAIT FOR NEXT POSITIVE
1662 004710 032777 000040 174674      BIT      #TM3,ANCSR ;ITERATION
1663 004716 001774      BEQ      .-6
1664 004720 013737 001644 001630      MOV      CLKCT,TEMPA ;STORE THIS TIME.
1665 004726 105737 001631      TSTB     TEMP+1
1666 004732 001005      BNE      4$      ;INSURE TIME IS ONE
1667 004734 123727 001630 000054      CMPB     TEMP,#54
1668 004742 103027      BHIS     TM3EXT   ;SECOND, PLUS OR MINUS
1669 004744 000410      BR       T3ERR   ;10 PER CENT
1670 004746 123727 001631 000001 4$:      CMPB     TEMP+1,#1
1671 004754 001004      BNE      T3ERR
1672 004756 123727 001630 000006      CMPB     TEMP,#6
1673 004764 103416      BLO      TM3EXT
1674 004766 005037 001124      T3ERR:   CLR      $GDDAT ;T3 CLOCK NOT WITH IN
1675 004772 005037 001126      CLR      $BDDAT
1676 004776 113737 001631 001124      MOVB     TEMP+1,$GDDAT ;TOLERANCE ISSUE T3
1677 005004 062737 000021 001126 1$:      ADD      #17,$BDDAT ;ADD 17 FOR EVERY CLOCK PULSE
1678 005012 105337 001630      DECB     TEMP
1679 005016 001372      BNE      1$
1680 005020 104017      ERROR   17
1681 005022 000240      TM3EXT:  NOP
1682
1683      ;*****
1684      ;ROUTINE TO READ THE STATIC CONDITION OF THE PANEL
1685      ;SWITCHES AND REPORT ANY SET TO THE 'ONE' STATE.
1686      ;*****
1687
1688      ;*****
1689 005024 000004      T$T13:   SCOPE
1690 005026 012737 000005 001160      MOV      #5,$TIMES ;;DO 5 ITERATIONS
1691 005034 012737 005042 001106      MOV      #SWSTAT,$LPADR ;;SET SCOPE LOOP ADDRESS
1692
1693 005042 017701 174550      SWSTAT:  MOV      @NPML,R1 ;READ THE PANEL
1694 005046 042701 177400      BIC      #177400,R1 ;STRIP THE LIGHT BITS
1695 005052 005701      TST      R1 ;ALL BITS SHOULD BE CLEAR.
1696 005054 001411      BEQ      SWEXT
1697 005056 013737 001616 001120      MOV      NPML,$GADR ;A BIT(S) SET REPORT
1698 005064 017737 174526 001126      MOV      @NPML,$BDDAT ;PANEL SWITCH ERROR
1699 005072 005037 001124      CLR      $GDDAT
1700 005076 104020      ERROR   20
1701 005100 000240      SWEXT:   NOP
1702
1703      ;*****
1704      ;ROUTINE TO INSURE THAT ALL DROUT FLOPS SET AND
1705      ;WILL BE RESET BY THE 16 M.S. TIMEOUT.
1706      ;*****
1707
1708      ;*****
1709 005102 000004      T$T14:   SCOPE
1710 005104 012737 000005 001160      MOV      #5,$TIMES ;;DO 5 ITERATIONS
1711 005112 012737 005120 001106      MOV      #CKDRT,$LPADR ;;SET SCOPE LOOP ADDRESS
1712
1713 005120 013737 001620 001120      CKDRT:  MOV      NTIM,$GADR
1714 005126 052777 000004 174456      BIS      #MAINT,ANCSR ;INSURE MAINT. MODE SET.
1715 005134 005037 001644      CLR      CLKCT ;CLEAR THE CLOCK COUNTER
1716 005140 052777 177400 174452      BIS      #177400,@NTIM ;SET ALL DROUT BITS

```

J03

```

1717 005146 022777 177400 174444      CMP      #177400,ANTIM  ;ALL SET?
1718 005154 001407                      BEQ      25          ;YES
1719 005156 017737 174436 001126      MOV      ANTIM,$BDDAT ;NO-ISSUE SET/RESET ERROR
1720 005164 012737 177404 001124      MOV      #177404,$GDDAT
1721 005172 104021                      ERROR    21
1722 005174 005777 174420          2$:      TST      ANTIM      ;HAS IM.S. RESET CLEARED BITS?
1723 005200 001412                      BEQ      DREXT     ;YES-EXIT
1724 005202 123727 001644 000002      CMPB    CLKCT,#2
1725 005210 103771                      BLO     25          ;NO-ISSLE SET/RESET ERROR.
1726 005212 017737 174402 001126      MOV      ANTIM,$BDDAT
1727 005220 005037 001124      CLR     $GDDAT
1728 005224 104021                      ERROR    21
1729 005226 032777 000010 174356  DREXT:  BIT      #BOOT,ANCSR ;CHECK IF A BOOT IN PROGRESS AND
1730 005234 001374                      BNE     -6          ;IF SO LOOP UNTIL FINISHED.
1731
1732
1733                                     ;;*****
1734                                     ;ROUTINE TO INSURE FROFLP (BIT0) CAN BE SET AND
1735                                     ;THAT IT WILL RESET WITH WDE NOT AND RESET..
1736                                     ;;*****
1737
1738                                     ;;*****
1739 005236 000004          TST15:  SCOPE
1740 005240 012737 000005 001160      MOV     #5,$TIMES   ;;DO 5 ITERATIONS
1741 005246 012737 005254 001106      MOV     #CKFLIT,$LPADR ;;SET SCOPE LOOP ADDRESS
1742
1743 005254 005037 001630          CKFLIT: CLR     TEMPA   ;CLEAR THE PASS COUNTER.
1744 005260 052777 000004 174324  FPASS:  BIS     #MAINT,ANCSR ;ENTER MAINTENANCE MODE.
1745 005266 052777 000002 174316      BIS     #WDE,ANCSR  ;CREATE AN ALARM AND FROFLP.
1746 005274 004037 011530          JSR     RD,FORCTA   ;GENERATE A CLEAR THEN A ALARM.
1747 005300 032777 000001 174312      BIT     #FROFLP,ANTIM ;FROFLP SET?
1748 005306 001007                      BNE     15          ;YES
1749 005310 017737 174304 001126      MOV     ANTIM,$BDDAT ;NO ISSUE FRO FLOP SET/RESET ERROR.
1750 005316 012737 000001 001124      MOV     #1,$GDDAT
1751 005324 104022                      ERROR    22
1752
1753 005326 005737 001630          1$:      TST     TEMPA      ;FIRST PASS?
1754 005332 001006                      BNE     35          ;NO
1755 005334 042777 000002 174250      BIC     #WDE,ANCSR  ;CLEAR ENABLE VIA WDE NOT
1756 005342 005237 001630          INC     TEMPA      ;YES -INCREMENT THE PASS COUNT.
1757 005346 000401                      BR      FRORST     ;GO CHECK FOR ERROR.
1758 005350 000005          3$:      RESET
1759 005352 032777 000001 174240  FRORST: BIT     #FROFLP,ANTIM ;TRY TO RESET FROFLP VIA RESET.
1760 005360 001406                      BEQ     FLTEXT     ;INSURE FROFLP CLEARS.
1761 005362 017737 174232 001126      MOV     ANTIM,$BDDAT ;WDE NOT DID NMT CLEAR
1762 005370 005037 001124      CLR     $GDDAT     ;FRO FLOP, ISSUE SET/RESET ERROR
1763 005374 104022                      ERROR    22
1764 005376 032777 001000 174206  FLTEXT: BIT     #TM2,ANCSR ;WAIT FOR T2 TO TIME OUT
1765 005404 001374                      BNE     -6
1766 005406 005737 001630          TST     TEMPA      ;FIRST PASS?
1767 005412 001722                      BEQ     FPASS     ;YES-CYCLE AGAIN AND USE RESET TO CLEAR.
1768
1769
1770                                     ;;*****
1771                                     ;INSURE MINUTE FLOPS (1,2,4,AND 8) CAN BE DIRECTLY
1772                                     ;SET VIA MA:NT MODE AND THE APPROPRIATE MINUTE

```

K03

```

1773                                     ;WRITE BIT. INSURE CLR1 RESETS ALL MINUTE FLOPS.
1774                                     ;;*****
1775                                     ;;*****
1776                                     ;;*****
1777 005414 000004 1ST16: SCOPE
1778 005416 012737 000005 001160      MOV      #5,$TIMES      ;;DO 5 ITERATIONS
1779 005424 012737 005432 001106      MOV      #CKMIN,$LPADR ;;SET SCOPE LOOP ADDRESS
1780
1781 005432 005077 174162      CKMIN: CLR      @NTIM
1782 005436 052777 000004 174146      BIS      @MAINT,@NCSR ;ENTER MAINT. MODE.
1783 005444 012703 000010      MOV      @MIN1,R3     ;PRE LOAD THE FIRST MIN. FLOP TO CHECK
1784 005450 004037 011530      MNST: JSR     RO,FORCTA ;GENERATE A CLEAR 1.
1785 005454 017702 174140      MOV      @NTIM,R2     ;INSURE ALL MINUTE FLOPS CLEAR
1786 005460 032702 000170      BIT      #170,R2
1787 005464 001406      BEQ      CKAMIN
1788 005466 010237 001126      MOV      R2,$BDDAT   ;NOT ALL CLEAR-ISSUE SET/
1789 005472 005037 001124      CLR      $GDDAT     ;RESET ERROR
1790 005476 104023      ERROR   23
1791 005500 000421      BR      MINEXT
1792 005502 004037 C11620      CKAMIN: JSR     RO,SYNC3 ;INSURE NO SPURIOUS RE-BOOT.
1793 005506 050377 174106      BIS      R3,@NTIM    ;SET THE MINUTE FLOP
1794 005512 030377 174102      BIT      R3,@NTIM    ;DID IT SET?
1795 005516 001006      BNE     15          ;YES-SET UP TO TRY NEXT BIT
1796 005520 017737 174074 001126      MOV      @NTIM,$BDDAT ;NO-ISSUE MINUTE FLOP
1797 005526 010337 001124      MOV      R3,$GDDAT   ;SET/RESET ERROR
1798 005532 104023      ERROR   23
1799 005534 006303      15:      ASL     R3          ;SET UP FOR NEXT MINUTE
1800 005536 032703 000200      BIT      #BIT07,R3   ;ARE WE DONE?
1801 005542 001742      BEQ     MNST        ;NO-TEST ANOTHER FLOP
1802 005544 004037 011530      MINEXT: JSR     RO,FORCTA ;GENERATE A CLEAR 1 PULSE
1803
1804
1805                                     ;;*****
1806                                     ;ROUTINE TO VERIFY OPERATION OF THE 15 SEC. FLOP IN
1807                                     ;MAINT. MODE. FLOP IS SET BY TOGGLING BIT01 16 TIMES.
1808                                     ;;*****
1809
1810                                     ;;*****
1811 005550 000004 1ST17: SCOPE
1812 005552 012737 000005 001160      MOV      #5,$TIMES      ;;DO 5 ITERATIONS
1813 005560 012737 005566 001106      MOV      #CK15,$LPADR ;;SET SCOPE LOOP ADDRESS
1814
1815 005566 005002      CK15: CLR      R2
1816 005570 005077 174024      CLR      @NTIM
1817 005574 052777 000004 174010      BIS      @MAINT,@NCSR ;ENTER MAINT. MODE.
1818 005602 004037 011620      JSR     RO,SYNC3     ;INSURE NO SPURIOUS RE-BOOT
1819 005606 012701 000020      MOV      #16,R1
1820 005612 004037 011466      JSR     RO,MSEC      ;GENERATE '16 SECONDS'
1821 005616 032777 000002 173774      BIT      #SEC15,@NTIM ;15 SEC. FLOP SET?
1822 005624 001010      BNE     RST15       ;YES-TRY TO RESET IT
1823 005626 017737 173766 001126      MOV      @NTIM,$BDDAT ;NO-ISSUE 'CAN'T SET 15 SEC. FLOP
1824 005634 012737 000002 001124      MOV      #2,$GDDAT   ;ERROR.
1825 005642 104024      ERROR   24
1826 005644 000417      BR      EXT15
1827 005646 012701 000017      RST15: MOV      #15,R1
1828 005652 004037 011466      JSR     RO,MSEC      ;GENERATE '15 SECONDS'

```

```

1829 005656 032777 000002 173734      BIT      #SEC15,ANTIM      ;DID '15 SEC.' FLOP RESET?
1830 005664 001407                      BEQ      EXT15          ;YES-EXIT
1831 005666 017737 173726 001126      MOV      ANTIM,$BDDAT   ;NO-ISSUE 'CAN'T RESET 15 SEC.
1832 005674 012737 000004 001124      MOV      #4,$GDDAT     ;FLOP' ERROR.
1833 005702 104025                      ERROR   25
1834 005704 004037 011530      EXT15: JSR      RO,FORCTA ;GENERATE A CLEAR 1 PULSE
1835
1836
1837                                     ;*****
1838                                     ;ROUTINE TO VERIFY OPERATION OF THE '30 SECOND' FLOP.
1839                                     ;SET INITIALLY WITH 31 "SECOND" PULSES AND RESET BY 31 + 30 'SECONDS'
1840                                     ;*****
1841
1842                                     ;*****
1843 005710 000004      †ST20: SCOPE
1844 005712 012737 000005 001160      MOV      #5,$TIMES     ;;DO 5 ITERATIONS
1845 005720 012737 005726 001106      MOV      #CK30,$LPADR  ;;SET SCOPE LOOP ADDRESS
1846
1847 005726 005077 173666      CK30: CLR      ANTIM
1848 005732 052777 000004 173652      BIS      #MAINT,ANCSR  ;ENTER MAINT. MODE.
1849 005740 004037 011620      JSR      RO,SYNC3      ;SYNC ON T3 HIGH.
1850 005744 012701 000037      MOV      #31,R1
1851 005750 004037 011466      JSR      RO,MSEC       ;GENERATE 31 'SECONDS'
1852 005754 032777 000004 173636      BIT      #SEC30,ANTIM  ;30 SEC. FLOP SET?
1853 005762 001010                      BNE     RST30          ;YES-GO TO RESET TEST
1854 005764 017737 173630 001126      MOV      ANTIM,$BDDAT
1855 005772 012737 000004 001124      MOV      #4,$GDDAT     ;NO-ISSUE 'CAN'T SET 30 SEC.
1856 006000 104026                      ERROR   26             ;FLOP' ERROR
1857 006002 000417                      BR      EXT30          ;AND EXIT
1858
1859 006004 012701 000036      RST30: MOV      #30,R1
1860 006010 004037 011466      JSR      RO,MSEC       ;GENERATE 30 'SECONDS'
1861 006014 032777 000004 173576      BIT      #SEC30,ANTIM  ;DID THIS CAUSE 30 SEC FLOP TO RESET?
1862 006022 001407                      BEQ     EXT30          ;YES
1863 006024 017737 173570 001126      MOV      ANTIM,$BDDAT  ;NO-ISSUE 'CAN'T RESET
1864 006032 012737 000002 001124      MOV      #2,$GDDAT     ;30 SEC. FLOP ERROR.
1865 006040 104027                      ERROR   27
1866
1867 006042 004037 011530      EXT30: JSR      RO,FORCTA ;GENERATE A CLEAR 1 PULSE
1868
1869
1870                                     ;*****
1871                                     ;ROUTINE TO INSURE CLR1 WILL RESET 15 A 30
1872                                     ;SECOND FLOPS
1873                                     ;*****
1874
1875                                     ;*****
1876 006046 000004      †ST21: SCOPE
1877 006050 012737 000005 001160      MOV      #5,$TIMES     ;;DO 5 ITERATIONS
1878 006056 012737 006064 001106      MOV      #CLRSEC,$LPADR ;;SET SCOPE LOOP ADDRESS
1879
1880 006064 004037 011720      CLRSEC: JSR      RO,GOCLK ;START THE REAL TIME CLOCK
1881 006070 005077 173524      CLR      ANTIM         ;CLEAR BEFORE CHACKING
1882 006074 052777 000004 173510      BIS      #MAINT,ANCSR  ;ENTER MAINT. MODE.
1883 006102 004037 011620      JSR      RO,SYNC3      ;SYNC ON T3 NOT-INSURE NO REBOOT.
1884 006106 012701 000001      SETA:  MOV      #1,R1

```

```

1885 006112 004037 011466 JSR RO,MSEC ;GENERATE A 'SECOND'
1886 006116 032777 000002 173474 BIT #SEC15,ANTIM
1887 006124 001770 BEQ SETA ;WAIT FOR '15 SEC' FLOP TO SET
1888 006126 004037 011530 JSR RO,FORCTA ;GENERATE A CLEAR 1
1889 006132 032777 000002 173460 BIT #SEC15,ANTIM ;FLOP NOW RESET?
1890 006140 001037 BNE SECERR ;NO-GO REPORT ERROR.
1891 006142 005077 173452 CLR ANTIM ;YES-SET UP TO CHECK 30 SEC. FLOP.
1892 006146 005037 001644 CLR CLKCT ;CLEAR REAL TIME CLOCK COUNTER.
1893 006152 012701 000001 SETB: MOV #1,R1
1894 006156 004037 011466 JSR RO,MSEC ;GENERATE A 'SECOND'.
1895 006162 123727 001644 000002 CMPB CLKCT,#2 ;ALLOW 16 MS FOR FLOP TO SET.
1896 006170 103005 BHIS 15 ;YES-ISSUE CANT SET 30 SEC FLOP.
1897 006172 032777 000004 173420 BIT #SEC30,ANTIM
1898 006200 001764 BEQ SETB ;WAIT FOR '30 SEC' FLOP TO SET.
1899 006202 000410 BR 25 ;IT SET-GO CLEAR IT.
1900 006204 012737 000004 001124 15: MOV #4,$GDDAT ;LOAD GOOD REG.
1901 006212 017737 173402 001126 MOV ANTIM,$BDDAT ;LOAD ACTUAL REG.
1902 006220 104026 ERROR 26 ;ISSUE CANT SET 30 SEC. FLOP ERROR.
1903 006222 000414 BR SECEXT ;AND EXIT.
1904 006224 004037 011530 25: JSR RO,FORCTA ;GENERATE A CLEAR 1 AND
1905 006230 032777 000004 173362 BIT #SEC30,ANTIM ;INSURE FLOP IS RESET
1906 006236 001406 BEQ SECEXT ;IT IS-EXIT
1907 006240 017737 173354 001126 SECERR: MOV ANTIM,$BDDAT ;ISSUE 'CAN'T RESET 15
1908 006246 005037 001124 CLR $GDDAT ;OR 30 SEC. FLOP' ERROR.
1909 006252 104030 ERROR 30
1910 006254 000240 SECEXT: NOP
1911
1912
1913 ;*****
1914 ;ROUTINE TO VERIFY PROPOGATION OF 15 SEC,30 SEC, AND
1915 ;1 THRU 8 MIN. FLOPS. ALL PROPOGATION ERRORS ARE REPORTED.
1916 ;AS ARE ALL CONFIGURATIONS WHICH CAUSE A REBOOT.
1917 ;*****
1918
1919 ;*****
1920 006256 000004 000001 001160 †ST22: SCOPE
1921 006260 012737 006274 001106 MOV #1,$TIMES ;;DO 1 ITERATION
1922 006266 012737 MOV #SERPRO,$LPADR ;;SET SCOPE LOOP ADDRESS
1923
1924 006274 052777 000004 173310 SERPRO: BIS #MAINT,ANC SR ;ENTER MAINT. MODE.
1925 006302 004037 011530 JSR RO,FORCTA ;GENERATE A CLEAR 1.
1926 006306 012703 000002 MOV #2,R3 ;R3 IS CHECK TIME.
1927 006312 005002 CLR R2 ;R2 IS TEST TIME.
1928
1929 006314 012701 000001 BTLP: MOV #1,R1 ;PRIME THE SECOND COUNTER.
1930 006320 004037 011466 JSR RO,MSEC
1931 006324 004037 011642 MINLP: JSR RO,SYNC3A ;SYNC ON TIME 3 TRUE.
1932 006330 010277 173264 MOV R2,ANTIM ;PARALLEL LOAD THE MIN. FLOPS.
1933 006334 012701 000017 MOV #15,R1 ;GENERATE 15 SECONDS.
1934 006340 004037 011466 JSR RO,MSEC
1935 006344 032777 000040 173240 BIT #TM3,ANC SR ;WAIT FOR TIME 3 NOT TO CAUSE REBOOT.
1936 006352 001374 BNE -6
1937 006354 000240 NOP ;ALLOW POSSIBLE RE-BOOT PROPOGATION.
1938 006356 017702 173236 MOV ANTIM,R2 ;GET CURRENT TIME.
1939 006362 001420 BEQ BOOTED ;TIME CLEAR -REBOOT OCCURRED.
1940 006364 042777 000001 173226 BIC #FROFLP,ANTIM ;CLEAR LOW ORDER BIT IF SET.

```

```

1941 006372 020203          CMP      R2,R3          ;PROPOGATION OK?
1942 006374 001405          BEQ      NWCNFG         ;YES-GET NEXT CONFIGURATION.
1943 006376 010337 001124    MOV      R3,$GDDAT     ;NO-ISSUE PROPOGATION ERROR.
1944 006402 010237 001126    MOV      R2,$BDDAT
1945 006406 104033          ERROR    33
1946
1947 006410 010302          NWCNFG: MOV     R3,R2          ;SET UP FOR NEXT INCREMENT.
1948 006412 005723          TST     (R3)+          ;INCREMENT MINUTE COUNT
1949 006414 032703 000200    BIT     #BIT07,R3      ;AT END OF MIN. FLOPS?
1950 006420 001047          BNE     MNXT           ;YES-EXIT TESTING FLOPS.
1951 006422 000740          BR      MINLP          ;NO-GO CHECK NEXT CONFIGURATION.
1952
1953 006424 010337 001624    BOOTED: MOV     R3,BTCND   ;STORE REBOOT CONFIGURATION.
1954 006430 010302          MOV     R3,R2          ;USE R2 AS WORK REG. FOR MIN.
1955 006432 104400 021261    TYPE    DBTCND         ;TYPE REBOOT TIME HEADER.
1956 006436 006202          ASR     R2
1957 006440 006202          ASR     R2              ;MOVE MINUTES INTO POSITION
1958 006442 006202          ASR     R2
1959 006444 010246          MOV     R2,-(SP)       ;SAVE R2 FOR TYPEOUT
1960
1961 006446 104404          TYPDS   ;TYPE MINUTES
1962 006450 005002          CLR     R2              ;GO TYPE--DECIMAL ASCII WITH SIGN
1963 006452 032737 000002 001624    BIT     #SEC15,BTCND   ;SET UP TO COUNT SECONDS.
1964 006460 001402          BEQ     1$              ;15 SEC. FLOP SET?
1965 006462 062702 000017 001624    ADD     #15.,R2         ;NO
1966 006466 032737 000004 001624 1$: BIT     #SEC30,BTCND   ;YES-ADD 15 SECONDS TO COUNT.
1967 006474 001402          BEQ     2$              ;30 SEC. FLOP SET?
1968 006476 062702 000036 2$: ADD     #30.,R2         ;NO
1969 006502          ;YES-ADD 30 SEC. TO COUNT.
1970 006502 010246          MOV     R2,-(SP)       ;SAVE R2 FOR TYPEOUT
1971
1972 006504 104404          TYPDS   ;TYPE SECONDS
1973 006506 104400 001165    TYPE    $SCRLF         ;GO TYPE--DECIMAL ASCII WITH SIGN
1974 006512 042703 000006    BIC     #6,R3           ;INSURE NEXT CONFIG. IS AN EVEN MINUTE.
1975 006516 062703 000010    ADD     #10,R3
1976 006522 012701 000001 3$: MOV     #1,R1          ;PRIME THE SECOND COUNTER.
1977 006526 004037 011466    JSR     R0,MSEC
1978 006532 004037 011642    JSR     R0,SYNC3A
1979 006536 000724          BR      NWCNFG         ;SYNC ON TIME 3 TRUE.
1980
1981 006540 122737 000070 001624 MNXT:  CMPB   #70,BTCND     ;TRY ANOTHER CONFIGURATION.
1982 006546 103004          BHIS   1$
1983 006550 012737 000100 001626    MOV     #100,NOBOOT    ;BOOT CONFIG. GREATER THAN 7 MIN?
1984 006556 000402          BR     2$              ;NO-SET NO BOOT CONFIG. TO 8 MIN.
1985 006560 005037 001626 1$: CLR     NOBOOT        ;YES-SET NO BOOT CONFIG. TO 0 MIN.
1986 006564 042777 000020 173020 2$: BIC     #BTENA,#NCSR  ;DISABLE SERIAL PROPOGATION.
1987
1988 ;*****
1989 ;ROUTINE TO INSURE TIME 3 CAN TOGGLE THE 15 SECOND
1990 ;COUNTER. COUNTER IS RUN TO A COUNT OF 14 IN MAINT.
1991 ;MODE, THEN MAINT MODE IS EXITED AND THE 15 SEC
1992 ;FLOP MUST SET WITHIN 3 SECONDS OR AN ERROR IS ISSUED.
1993 ;*****
1994
1995 ;*****
1996 †ST23: SCOPE

```

```

1997 006574 012737 000005 001160      MOV      #5,STIMES      ;;DO 5 ITERATIONS
1998 006602 012737 006610 001106      MOV      #SECNTR,$LPADR ;;SET SCOPE LOOP ADDRESS
1999
2000 006610 004037 011720      SECNTR: JSR      RD,GOCLK      ;START THE CLOCK.
2001 006614 052777 000004 172770      BIS      #MAINT,$NCSR    ;ENTER MAINT. MODE
2002 006622 004037 011534      JSR      RD,FORCT      ;CLEAR FLOPS AND COUNTER
2003 006626 013777 001626 172764      MOV      NCB00T,$NTIM   ;INSURE NO REBOOTS
2004 006634 012701 000016      MOV      #14,$R1
2005 006640 004037 011466      JSR      RD,MSEC      ;GENERATE 14 'SECONDS'
2006 006644 032777 000010 172740      BIT      #BOOT,$NCSR    ;LOOP UNTIL BOOT GENERATED BY MSEC
2007 006652 001374      BNE      #-6          ;IS THRU.
2008 006654 005037 001644      CLR      CLKCT
2009 006660 042777 000024 172724      BIC      #MAINT+$BTENA,$NCSR ;EXIT MAINT. MODE.
2010 006666 032777 000002 172724      IS:     BIT      #SEC15,$NTIM ;'15 SEC' FLOP SET?
2011 006674 001013      BNE      CNTRXT      ;YES-EXIT
2012 006676 123727 001645 000003      CMPB    CLKCT+1,#3   ;NO-HAVE WE COUNTED 3 REAL SECONDS?
2013 006704 103770      BLO      IS          ;NO-KEEP LOOKING
2014 006706 017737 172706 001126      MOV      $NTIM,$B00AT  ;YES-ISSUE 'TIME 3 DOES NOT
2015 006714 012737 000002 001124      MOV      #2,$G00AT    ;TOGGLE 15 SEC. COUNTER' ERROR.
2016 006722 104031      ERROR    31
2017 006724 052777 000004 172660      CNTRXT: BIS      #MAINT,$NCSR ;RE-ENTER MAINT. MODE
2018
2019      ;;*****
2020      ;ROUTINE TO CREATE A 'BOOT(H)' BY: A.- KEEPING REBOOT
2021      ;LOW AND FORCING BTENA, AND B. GENERATING 15 "SECONDS" UNTIL
2022      ;BOOT CONFIGURATION IS GENERATED.
2023      ;;*****
2024
2025      ;;*****
2026      $TST24: SCOPE
2027 006732 000004      MOV      #5,STIMES      ;;DO 5 ITERATIONS
2028 006734 012737 000005 001160      MOV      #SETBT,$LPADR ;;SET SCOPE LOOP ADDRESS
2029 006742 012737 006750 001106
2030 006750 013737 001612 001120      SETBT:  MOV      NCSR,$G0ADR ;SET UP REG. ADD. FOR PRINTOUT.
2031 006756 052777 000004 172626      BIS      #MAINT,$NCSR    ;ENTER MAINT. MODE
2032 006764 004037 011530      JSR      RD,FORCTA      ;GENERATE A CLR 1.
2033 006770 004037 011620      JSR      RD,SYNC3      ;TEST ON TIME 3 LOW.-NO REBOOT.
2034 006774 004037 011720      JSR      RD,GOCLK      ;START THE CLOCK.
2035 007000 052777 000020 172604      BIS      #BTENA,$NCSR   ;SET BOOT ENABLE
2036 007006 017702 172600      MOV      $NCSR,$R2     ;READ REG D.
2037 007012 032702 000010      BIT      #BOOT,$R2     ;WAS 'BOOT' (BIT3) SET?
2038 007016 001006      BNE      SET2          ;YES-GO TO NEXT SET ATTEMPT.
2039 007020 010237 001126      MOV      $R2,$B00AT
2040 007024 012737 042236 001124      MOV      #42236,$G00AT ;NO-ISSUE 'CAN'T SET BOOT'
2041 007032 104032      ERROR    32          ;ERROR
2042 007034 032777 000010 172550      SET2:   BIT      #BOOT,$NCSR ;BOOT SET?
2043 007042 001413      BEQ      SET2A        ;NO
2044 007044 123727 001645 000002      CMPB    CLKCT+1,#2   ;YES-TIME EQUAL 2 SEC?
2045 007052 001370      BNE      SET2          ;NO-LOOP
2046 007054 017737 172532 001126      MOV      $NCSR,$B00AT  ;YES-LOAD AND ISSUE 'BOOTDID NOT RESET'
2047 007062 012737 042224 001124      MOV      #42224,$G00AT
2048 007070 104036      ERROR    36
2049 007072 004037 011530      SET2A:  JSR      RD,FORCTA      ;CLEAR BOOT ENABLE.
2050 007076 004037 011642      JSR      RD,SYNC3A     ;TEST ON TIME 3 HIGH.-ALLOW REBOOT.
2051 007102 052777 000002 172502      BIS      #W0E,$NCSR    ;SET MAINT. W0E.
2052 007110 012701 000015      IS:     MOV      #15,$R1  ;GENERATE 15 'SECOND'
2053 007114 004037 011466      JSR      RD,MSEC      ;MINUTE CONFIG. PLUS SEC. =BOOT.

```

```

2053 007120 017702 172466      MOV      QNCSR,R2      ;GET REG. 0
2054 007124 032702 000010      BIT      #BOOT,R2      ;BOOT SET?
2055 007130 001012                BNE      SETBXT        ;YES-EXIT
2056 007132 122777 000176 172460      CMPB     #176,QNTIM     ;COUNTED TO MAXIMUM?
2057 007140 101363                BHI      IS            ;NO-KEEP TRYING.
2058 007142 010237 001126      MOV      R2,$BODAT     ;YES-ISSUE 'CAN'T SET BOOT'
2059 007146 012737 042234 001124      MOV      #42234,$GDDAT ;ERROR.
2060 007154 104032                ERROR    32
2061 007156 022737 000001 001104  SETBXT:  CMP      #1,$ICNT      ;REPORT ALARM BOOT ON 1ST ITERATION.
2062 007164 001021                BNE      3S
2063 007166 005037 001644      CLR      CLKCT         ;CLEAR THE REAL TIME COUNT.
2064 007172 004037 011534      JSR      RD,FORCT      ;SUPPORT THE T1,T2 SEQUENCE
2065 007176 032777 000010 172406  1S:     BIT      #BOOT,QNCSR   ;BOOT GENERATED?
2066 007204 001007                BNE      2S            ;YES,EXIT
2067 007206 123727 001645 000002      CMPB     CLKCT+1,#2    ;ALLOW 2 SEC FOR BOOT.
2068 007214 103770                BLO      IS
2069 007216 104400 021337      TYPE     ,DALNBT      ;DISPLAY NO ALARM BOOT.
2070 007222 000402                BR       3S
2071 007224 104400 021436 2S:     TYPE     ,DALBT       ;DISPLAY ALARM CAUSED A BOOT.
2072 007230 042777 000002 172354  3S:     BIC      #WDE,QNCSR    ;CLEAR WATCH DOG ENABLE -EXIT.
2073
2074      ;*****
2075      ;ROUTINE TO INSURE T1 NOT WILL INTERRUPT TO MW11 VECTOR +4
2076      ;WHEN THE CONDITIONS T1 NOT AND MAINT. AND BIT9,REG4 ARE MET.
2077      ;*****
2078
2079      ;*****
2080 007236 000004      TST2S:  SCOPE
2081 007240 012737 000005 001160      MOV      #5,$TIMES    ;DO 5 ITERATIONS
2082 007246 012737 007254 001106      MOV      #TSTINT,$LPADR ;SET SCOPE LOOP ADDRESS
2083
2084 007254 005077 172336      TSTINT:  CLR      QNPNL      ;INSURE NO SPURIOUS INT.
2085 007260 013701 001636      MOV      #TINT,R1
2086 007264 012721 007366      MOV      #INTCK,(R1)+ ;LOAD TIME 1 INT. AND VECTOR.
2087 007270 012711 000240      MOV      #240,(R1)    ;SET INT. LEVEL TO 5. TO ALLOW CLOCK INT.
2088
2089 007274 004037 011720      JSR      RD,GOCLK     ;START THE CLOCK
2090 007300 052777 000004 172304      BIS      #MAINT,QNCSR  ;ENTER MAINT. MODE
2091 007306 052777 001000 172302      BIS      #T1INT,QNPNL ;AND ALLOW TIME 1 INT.
2092 007314 052777 000001 172270      BIS      #TGO,QNCSR   ;CREATE A TIME1
2093 007322 123727 001644 000003      CMPB     CLKCT,#3     ;ALLOW 48 M.S. FOR INT.
2094 007330 103774                BLO      -6           ;THEN ISSUE NO TIME 1 INT.
2095 007332 042777 001000 172256      BIC      #T1INT,QNPNL ;RESET THE INT. ENABLE.
2096 007340 013737 001616 001120      MOV      #NPNL,$GADR  ;ERROR MESSAGE.
2097 007346 017737 172244 001126      MOV      #NPNL,$BODAT
2098 007354 012737 001000 001124      MOV      #1000,$GDDAT
2099 007362 104037                ERROR    37
2100 007364 000404                BR       INTCKA
2101
2102 007366 032626      INTCK:  BIT      (R6)+(R6)+ ;POP TE OLD ADDRESS AND PSW
2103 007370 042777 001000 172220      BIC      #T1INT,QNPNL ;ALLOW NO MORE T-1 INT.
2104 007376 013701 001636      INTCKA:  MOV      #TINT,R1 ;RESTORE VECTOR AND PSW ADDRESSES
2105 007402 012721 000004      MOV      #4,(R1)+    ;TO TRAP CONDITIONS.
2106 007406 005011                CLR      (R1)
2107 007410 005046                CLR      -(R6)
2108 007412 012746 007420      MOV      #IS,-(R6)   ;CLEAR THE PSW LSI 11 STYLE.

```

```

2109 007416 000002          RTI
2110 007420 000240          1$:  NOP
2111
2112 ;*****
2113 ;ROUTINE TO INSURE T1-T2 SEQUENCE WILL CREATE A G-LITE
2114 ;AND EXTINGUISH THE R-LINE. T-1 SUPPORT IS THEN DROPPED
2115 ;AND THE G-LITE SHOULD BE OUT AND R-LITE PLUS ALARM
2116 ;ON. THE PROCEDURE IS REPEATED FOR T-2.
2117 ;*****
2118
2119 ;*****
2120 007422 000004          TST26: SCOPE
2121 007424 012737 000005 001160      MOV      #5, $TIMES          ;; DO 5 ITERATIONS
2122 007432 012737 007440 001106      MOV      #SUP12, $LPADR     ;; SET SCOPE LOOP ADDRESS
2123
2124 007440 013737 001612 001120      SUP12:  MOV      NCSR, $GADR   ; LOAD ADDRESS FOR POSSIBLE ERROR
2125 007446 052777 000004 172136      BIS      #MAINT, @NCSR     ; ENTER MAINT MODE-INSURE NO REBOOT.
2126 007454 004037 011534              JSR      RO, FORCT         ; SUPPORT T1-T2 FOR COUNT OF 5.
2127 007460 000240              NOP
2128 007462 032777 040200 172122      BIT      #40200, @NCSR    ; ARE 'R-LITE' AND 'ALARM' CLEAR?
2129 007470 001407              BEQ      1$,              ; YES-EXIT
2130 007472 017737 172114 001126      MOV      @NCSR, $BDDAT
2131 007500 012737 031014 001124      MOV      #31014, $GDDAT
2132 007506 104006              ERROR   6
2133 007510 004037 011534          1$:  JSR      RO, FORCT         ; NO ISSUE 'CAN'T RESET ALARM'
2134 007514 000240              NOP                          ; ERROR.
2135 007516 052777 000001 172066      BIS      #TGO, @NCSR     ; GENERATE A G-LITE
2136 007524 052777 000001 172060      BIS      #TGO, @NCSR     ; ALLOW FOR GATE SETTLING
2137 007532 032777 000200 172052      BIT      #ALARM, @NCSR   ; FORCE AN EXTRA T-1.
2138 007540 001007              BNE     2$,              ; FORCE AN EXTRA T-1.
2139 007542 017737 172044 001126      MOV      @NCSR, $BDDAT   ; ALARM CONDITION?
2140 007550 012737 041604 001124      MOV      #41604, $GDDAT  ; YES-EXIT TEST
2141 007556 104034              ERROR   34
2142 007560 004037 011534          2$:  JSR      RO, FORCT         ; NO-ISSUE 'CAN'T FORCE ALARM'
2143 007564 032777 001000 172020      BIT      #T42, @NCSR    ; ERROR
2144 007572 001374              BNE     .-6,             ; SET UP G-LITE CONDITION.
2145 007574 000240              NOP                          ; NOW WAIT FOR T-2 TO
2146 007576 032777 000200 172006      BIT      #ALARM, @NCSR   ; TIME OUT.
2147 007604 001007              BNE     SUPXT,          ; ALARM CONDITION?
2148 007606 017737 172000 001126      MOV      @NCSR, $BDDAT   ; YES-EXIT TEST
2149 007614 012737 040604 001124      MOV      #40604, $GDDAT
2150 007622 104005              ERROR   5
2151 007624 032777 002000 171306      SUPXT:  BIT      #SW10, @SWR ; NO ISSUE 'CAN'T FORCE ALARM'
2152 007632 001414              BEQ     ALTST,          ; ERROR.
2153 007634 005737 001100              TST     $PASS           ; LOOP ON CYLIC TESTS?
2154 007640 001404              BEQ     1$,              ; NO-CONTINUE WITH INTERVENTION TESTS
2155 007642 023727 001104 000004      CMP     $ICNT, #4        ; FIRST PASS?
2156 007650 101403              BLOS   2$,              ; YES-REPORT CYCLE.
2157 007652 104400 022345          1$:  TYPE  ,DEND           ; NO-BUT LAST ITERATION?
2158 007656 000005              RESET ; NO-DO NOT REPORT CYCLE OR RESET SYS.
2159 007660 000137 002670          2$:  JMP     TSTLOP         ; YES-TYPE END- AND LOOP
2160 007664 000240          ALTST: NOP
2161
2162 ;*****
2163 ;ROUTINE TO INSURE PANEL CAN BE ENABLED. OPERATOR
2164

```

```

2165 ; IS ALLOWED ONE MINUTE TO COMPLETE OPERATION BEFORE
2166 ; ERROR IS REPORTED.
2167 ;;*****
2168
2169 ;;*****
2170 007666 000004          TST27: SCOPE
2171 007670 012737 000001 001160      MOV      #1,STIMES      ;; DO 1 ITERATION
2172 007676 012737 007704 001106      MOV      #CKPNLK,$LPADR ;; SET SCOPE LOOP ADDRESS
2173
2174 007704 104400 020246          CKPNLK: TYPE      ,DPNLK1      ; TELL OPERATOR TO ENABLE THE PANEL
2175 007710 104400 020222          TYPE      ,DCRCNT
2176 007714 004037 011720          JSR      RD,GOCLK      ; START THE CLOCK.
2177 007720 005037 001644          LPNLK: CLR      CLKCT      ; CLEAR CLOCK COUNT
2178 007724 032777 000001 171664      IS:      BIT      #BIT00,$PNL  ; PANEL ENABLED?
2179 007732 001013          BNE      PNLKXT      ; YES-EXIT "EST
2180 007734 004037 011664          JSR      RD,CKCRT      ; RECEIVED CARRIAGE RETURN?
2181 007740 000401          BR      2$
2182 007742 000407          BR      PNLKXT      ; YES-EXIT.
2183 007744 123727 001645 000036      2$:      CMPB     CLKCT+1,#30.  ; NO-30 SECONDS ELAPSED?
2184 007752 103764          BLO      IS          ; NO-KEEP LOOKING
2185 007754 104400 020405          TYPE      ,DPLKER      ; YES-ISSUE PANEL NOT LOCKED.
2186 007760 000757          BR      LPNLK      ; NO-LOOP ON TEST
2187 007762 000240          PNLKXT: NOP
2188
2189 ;;*****
2190 ; ROUTINE TO ALLOW/REIND OPERATOR TO TEST LAMPS VIA
2191 ; LAMP TEST SWITCH. A CARRIAGE RETURN WILL CAUSE LAMPS
2192 ; TO REFLECT CONDITION OF THE SWITCHES DYNAMICALLY.
2193 ; CARRIAGE RETURN TO EXIT.
2194 ;;*****
2195
2196 ;;*****
2197 007764 000004          TST30: SCOPE
2198 007766 012737 000001 001160      MOV      #1,STIMES      ;; DO 1 ITERATION
2199 007774 012737 010002 001106      MOV      #SALMP,$LPADR ;; SET SCOPE LOOP ADDRESS
2200
2201 010002 005077 171610          SALMP: CLR      $PNL      ; CLEAR ANDY PANEL FLOPS.
2202 010006 104400 020105          TYPE      ,DLMPTS      ; DISPLAY LAMP TEST MESSAGE
2203 010012 104400 020222          TYPE      ,DCRCNT
2204 010016 004037 011664          JSR      RD,CKCRT      ; AND WAIT FOR CARRIAGE RET.
2205 010022 000775          BR      -4
2206
2207 010024 104400 020164          LPNL:   TYPE      ,DLPSW      ; TYPE LAMPS = SWITCHES
2208 010030 017701 171562          MOV      $PNL,R1      ; GET THE SWITCHES.
2209 010034 005077 171556          CLR      $PNL      ; CLEAR OLD CONFIGURATION
2210 010040 000301          SWAB     R1          ; MOVE SWITCHES TO LAMPS AND
2211 010042 010177 171550          MOV      R1,$PNL      ; DISPLAY THEM
2212 010046 004037 011664          JSR      RD,CKCRT      ; RECEIVED CARRIAGE RETURN?
2213 010052 000766          BR      LPNL      ; NO-LOOP
2214
2215 ;;*****
2216 ; ROUTINE TO INSURE DEPRESSING PANEL INTERRUPT LUTTON
2217 ; CAUSES AN INTERRUPT
2218 ;;*****
2219
2220 ;;*****

```

```

2221 010054 000004          TST31: SCOPE
2222 010056 012737 000001 001160      MOV      #1,$TIMES      ;;DO 1 ITERATION
2223 010064 012737 010072 001106      MOV      #PNLINT,$LPADR ;;SET SCOPE LOOP ADDRESS
2224
2225 010072 013701 001634          PNLINT: MOV      NWVECT,R1      ;GET THE VECTOR AND
2226 010076 012721 010214          MOV      #INTIN,(R1)+   ;LOAD THE INTERRUPT ADDRESS
2227 010102 012711 000240          MOV      #240,(R1)     ;AND PSW
2228 010106 005037 001630          CLR      TEMPA         ;CLEAR THE INTERRUPT COUNTER.
2229 010112 004037 011720          JSR      RD,GOCLK      ;START THE CLOCK.
2230
2231 010116 104400 020434          TYPE     ,DPNINT       ;DISPLAY 'DEPRESS INT. BUTTON'.
2232 010122 104400 020222          TYPE     ,DCRCNT
2233 010126 005037 001644          CLR      CLKCT        ;CLEAR CLOCK COUNT
2234 010132 052777 000100 171452  INTCNT: BIS      #INTENA,#NCSR ;ENABLE THE PANEL INTERRUPTS.
2235 010140 005046          CLR      -(R6)
2236 010142 012746 010150          MOV      #15,-(R6)    ;CLEAR THE PSW LSI 11 STYLE.
2237 010146 000002          RTI
2238 010150 004037 011664          15:      JSR      RD,CKCRT ;LOOK FOR A CARRIAGE RET.
2239 010154 000401          BR       10$          ;NO C/R-CONTINUE
2240 010156 000422          BR       CREXT
2241 010160 123727 001645 000017  10$:     CMPB     CLKCT+1,#15.   ;ALLOW .25 MINUTE FOR BUTTON
2242 010166 103770          BLO     15           ;TO BE DEPRESSED.
2243 010170 042777 000100 171414          BIC      #INTENA,#NCSR ;CLEAR THE ENABLE BIT
2244 010176 005737 001630          TST     TEMPA        ;ALLREADY STORED A INT.?
2245 010202 001401          BEQ     2$           ;NO-PRINT ERROR
2246 010204 000407          BR       CREXT       ;YES-JUST COUNT IT.
2247 010206 104400 020521          2$:      TYPE     ,DNOINT ;THEN TYPE ERROR AND
2248 010212 000404          BR       CREXT       ;EXIT TEST.
2249 010214 032626          INTIN:  BIT      (R6)+,(R6)+ ;POP THE OLD ADDRESS AND PSW
2250 010216 005237 001630          INC     TEMPA        ;INCREMENT THE INT. COUNT.
2251 010222 000743          BR       INTCNT      ;AND GO LOOK FOR ANOTHER.
2252 010224 042777 000100 171360  CREXT:  BIC      #INTENA,#NCSR ;CLEAR THE ENABLE BIT.
2253 010232 013701 001634          MOV     NWVECT,#1
2254 010236 012721 000004          MOV     #4,(R1)+     ;RESTORE THE TRAP
2255 010242 005011          CLR     (R1)
2256 010244 005337 001630          DEC     TEMPA        ;CHECK FOR ONLY ONE INT.
2257 010250 001412          BEQ     PNLEXT       ;YES-GO EXIT
2258 010252 005237 001530          INC     TEMPA        ;MULTIPLE INT. DISPLAY ERROR AND COUNT.
2259 010256 001407          BEQ     PNLEXT       ;INT. COUNT = 0, BYPASS MULTIPLE ERROR PRINTOUT.
2260 010260 104400 020575          TYPE     ,DMULNT
2261 010264 013746 001630          MOV     TEMPA,-(SP)  ;;SAVE TEMPA FOR TYPEOUT
2262 010270 104404          TYPDS          ;;GO TYPE--DECIMAL ASCII WITH SIGN
2263 010272 104400 001165          TYPE     ,SCLRF
2264 010276 000240          PNLEXT: NOP
2265
2266
2267 ;;*****
2268 ;ROUTINE TO INSURE WOE CAN BE SET AND RESET
2269 ;FROM THE PANEL SWITCH. A SUBTEST WILL INSURE
2270 ;THAT RESET WILL CLEAR THE FRO FLOP.
2271 ;;*****
2272
2273 ..*****
2274 010300 000004          †TST32: SCOPE
2275 010302 012737 000001 001160      MOV      #1,$TIMES      ;;DO 1 ITERATION
2276 010310 012737 010316 001106      MOV      #W0ECK,$LPADR ;;SET SCOPE LOOP ADDRESS

```

```

2277
2278 010316 005037 001644      WDECK: CLR      CLKCT      ;CLEAR THE CLOCK COUNT.
2279 010322 104400 020325      TYPE      DPNLK      ;ISSUE -TURN PANEL LOCK TO OFF.
2280 010326 032777 000001 171262 2S:  BIT      #PNLCK,#PNPL ;PANEL LOCK OFF?
2281 010334 001405      BEQ      3S          ;YES-EXIT
2282 010336 123727 001645 000012      CMPB     CLKCT+1,#10. ;ALLOW 10 SEC. AND RE ISSUE MESSAGE.
2283 010344 103770      BLO      2S
2284 010346 000763      BR       WDECK
2285 010350 004037 011530      JSR      RD,FORCTA   ;GENERATE A CLEAR
2286 010354 104400 020666      TYPE      ,DWDEON   ;ISSUE 'TURN WDE TO ON'.
2287 010360 104400 020222      TYPE      ,DCRCNT
2288 010364 004037 011720      JSR      RD,GCLK
2289 010370 005037 001644      WDELP: CLR      CLKCT      ;START THE CLOCK.
2290 010374 032777 000002 171210 1S:  BIT      #WDE,#NCSR  ;CLEAR CLOCK COUNT AND START
2291 010402 001013      BNE      CKWDOF     ;WATCH DOG ENABLE 'ON'?
2292 010404 004037 011664      JSR      RD,CKCRT   ;YES-GO SHUT IT OFF.
2293 010410 000401      BR       2S         ;RECEIVED CARRIAGE RETURN?
2294 010412 000450      BR       WDEXT
2295 010414 123727 001645 000036 2S:  CMPB     CLKCT+1,#30. ;YES-EXIT
2296 010422 103764      BLO      1S         ;ELAPSED TIME EQUAL 1 MIN.?
2297 010424 104400 021522      TYPE      ,DNOWDE   ;NO-LOOP
2298 010430 000757      BR       WDELP     ;ISSUE 'NO WATCH DOG ENABLE'
2299 010432 104400 020747      TYPE      ,DWDEOF   ;LOOP
2300 010436 005037 001644      WDOFLP: CLR      CLKCT      ;ISSUE 'TURN WDE OFF'
2301 010442 032777 000002 171142 1S:  BIT      #WDE,#NCSR  ;WDE NOW CLEARED?
2302 010450 001412      BEQ      CKWRST     ;YES-GO RESET FRO
2303 010452 123727 001645 000036      CMPB     CLKCT+1,#30. ;NO-ELAPSED TIME EQUAL 1 MIN.?
2304 010460 103770      BLO      1S         ;NO-LOOP
2305 010462 104400 021572      TYPE      ,DALWDE   ;YES-ISSUE 'WDE NOT RESET'.
2306 010466 004037 011664      JSR      RD,CKCRT   ;RECEIVED A C/R?
2307 010472 000761      BR       WDOFLP    ;NO-LOOP
2308 010474 000417      BR       WDEXT     ;YES-EXIT
2309 010476 000005      CKWRST: RESET      ;INITIALIZE THE SYSTEM
2310 010500 032777 000001 171112      BIT      #FROFLP,#NTIM ;FRO FLOP RESET?
2311 010506 001412      BEQ      WDEXT     ;YES-EXIT TEST.
2312 010510 013737 001612 001120      MOV      NCSR,$GDADR
2313 010516 017737 171070 001126      MOV      #NCSR,$BDDAT
2314 010524 012737 040200 001124      MOV      #40200,$GLDAT ;NO-LOAD AND USSUE 'CAN'T
2315 010532 104035      ERROR    3S        ;RESET FRO FLOP' ERROR.
2316 010534 000240      WDEXT:  NOP
2317
2318 ;*****
2319 ;ROUTINE TO INSURE 'PANEL BOOT' WILL CAUSE A BOOT
2320 ;ENABLE (REG0, BIT4) AND THAT GENERATING A BOOT
2321 ;CLEAR WILL CLEAR THE BOOT ENABLE.
2322 ;*****
2323
2324 ;*****
2325 †ST33: SCOPE
2326 010536 000004      MOV      #1,$TIMES  ;;DO 1 ITERATION
2327 010540 012737 000001 001160      MOV      #CKBTEN,$LPADR ;;SET SCOPE LOOP ADDRESS
2328 010546 012737 010554 001106
2329 010554 005077 171032      CKBTEN: CLR      #NCSR
2330 010560 052777 000004 171024      BIS      #MAINT,#NCSR
2331 010566 004037 011530      JSR      RD,FORCTA  ;GENERATE A BOOT CLEAR
2332 010572 104400 021646      TYPE      ,DBTENA   ;ISSUE 'PRESS PANEL BOOT'.

```

```

2333 010576 004037 011720          JSR      RO,GOCLK          ;START THE CLOCK.
2334 010602 005037 001644          CLR      CLKCT            ;CLEAR CLOCK COUNT AND START
2335 010606 032777 000020 170776 LPBTEN: BIT      #BTENA,#NCSR    ;RECEIVED A 'BOOT ENABLE'?
2336 010614 001014          BNE      RSTENA          ;YES-GO TRY TO RESET IT
2337 010616 004037 011664          JSR      RO,CKCRT        ;RECEIVED A C/R?
2338 010622 000401          BR       2$              ;NO-CONTINUE
2339 010624 000436          BR       BTENXT          ;YES-EXIT TEST.
2340 010626 123727 001645 000036 2$:  CMPB    CLKCT+1,#30.    ;ELAPSED TIME EQUAL .5 MIN.?
2341 010634 103764          BLO     1$              ;NO-LOOP
2342 010636 104400 021772          TYPE    ,DNOPNB         ;YES-ISSUE 'NO BOOT ENA.' MESSAGE
2343 010642 000757          BR      LPBTEN          ;NO-LOOP.
2344 010644 000426          BR      BTENXT          ;YES-EXIT TEST.
2345 010646 005037 001644          RSTENA: CLR     CLKCT
2346 010652 123727 001645 000002  CMPB    CLKCT+1,#2      ;ALLOW 2 SECONDS FOR THE CLUB TO
2347 010660 001374          BNE     -6              ;TO REMOVE HIS FINGER FROM BUTTON.
2348 010662 004037 011530          JSR     RO,FORCTA       ;GENERATE A BOOT CLEAR
2349 010666 032777 000020 170716  BIT     #BTENA,#NCSR    ;'BOOT ENABLE' NOW CLEARED?
2350 010674 001412          BEQ     BTENXT          ;YES-EXIT
2351 010676 013737 001612 001120  MOV     NCSR,$GADR
2352 010704 017737 170702 001126  MOV     #NCSR,$BODAT    ;NO-LOAD AND USSUE 'CAN'T
2353 010712 012737 040224 001124  MOV     #40224,$GODAT   ;RESET BOOT ENA.' ERROR.
2354 010720 104012          ERROR   12
2355
2356 010722 104400 020747          BTENXT: TYPE    ,DWDEF    ;TURN WATCH DOG ENA. TO OFF.
2357
2358          ;;*****
2359          ;ROUTINE TO ALLOW 'SCOPING' OF THE FIELD RELAY
2360          ;CONTACTS. THE RELAY WILL BE 'ON' 5 SEC, OFF 5 SEC, ETC.,
2361          ;UNTIL A C/R IS SEEN.
2362          ;;*****
2363
2364          ;;*****
2365 010726 000004          †ST34: SCOPE
2366 010730 012737 000001 001160  MOV     #1,$TIMES      ;;DO 1 ITERATION
2367 010736 012737 010744 001106  MOV     #CYCFRO,$LPADR ;;SET SCOPE LOOP ADDRESS
2368
2369 010744 052777 000004 170640  CYCFRO: BIS     #MAINT,#NCSR ;INSURE MAINT. MODE SET.
2370 010752 104400 022062          TYPE    ,DCYCFR
2371 010756 004037 011720          JSR     RO,GOCLK        ;START THE CLOCK.
2372 010762 052777 000002 170622  SETLP: BIS     #WDE,#NCSR ;FORCE A WDE.
2373 010770 004037 011530          RSETLP: JSR    RO,FORCTA ;GENERATE A CLEAR AND ALARM.
2374 010774 005037 001644          CLR     CLKCT          ;CLEAR CLOCK COUNT AND INSURE
2375 011000 123727 001645 000005  CMPB    CLKCT+1,#5     ;ELAPSED TIME EQUAL 5 SEC.?
2376 011006 103774          BLO     -6              ;NO-LOOP
2377 011010 004037 011664          JSR     RO,CKCRT        ;RECEIVED A C/R?
2378 011014 000401          BR      1$              ;NO
2379 011016 000410          BR      CYCFXT          ;YES-EXIT
2380 011020 032777 000002 170564  1$:  BIT     #WDE,#NCSR    ;IF WDE IS SET, CLEAR IT
2381 011026 001755          BEQ     SETLP          ;OR VICE-VERSA
2382 011030 042777 000002 170554  BIC     #WDE,#NCSR
2383 011036 000754          BR      RSETLP
2384 011040 005077 170546          CYCFXT: CLR     #NCSR
2385
2386          ;;*****
2387          ;ROUTINE TO 'RIPPLE DROUT BITS AT 1 SECOND INTERVALS.
2388          ;;*****

```



```

2445
2446 011264
2447 011264 000004
2448 011266 005037 001102
2449 011272 005037 001160
2450 011276 005237 001100
2451 011302 042737 100000 001100
2452 011310 005327
2453 011312 000001
2454 011314 003022
2455 011316 012737
2456 011320 000001
2457 011322 011312
2458 011324 104400 011371
2459 011330 013746 001100
2460 011334 104404
2461 011336 104400 011366
2462 011342 013700 000042
2463 011346 001405
2464 011350 000005
2465 011352 004710
2466 011354 000240
2467 011356 000240
2468 011360 000240
2469 011362
2470 011362 000137
2471 011364 002270
2472 011366 377 377 000
2473 011371 015 042412 042116
2474 011376 050040 051501 020123
2475 011404 000043
2476 011406 000000
2477
2478
2479
2480
2481
2482
2483
2484 011410 105237 001644
2485 011414 123727 001644 000074
2486 011422 001004
2487 011424 105037 001644
2488 011430 105237 001645
2489 011434 042777 000200 170170
2490 011442 000002
2491
2492
2493
2494
2495
2496
2497 011444 005037 001120
2498 011450 005037 001122
2499 011454 005037 001124
2500 011460 005037 001126

$EOP:
SCOPE
CLR $STNM ;; ZERO THE TEST NUMBER
CLR $TIMES ;; ZERO THE NUMBER OF ITERATIONS
INC $PASS ;; INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;; DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;; LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;; YES
MOV (PC)+,2(PC)+ ;; RESTORE COUNTER
$ENDCT: .WORD 1
$EOPCT
TYPE $SENDMG ;; TYPE "END PASS #"
MOV $PASS,-(SP) ;; SAVE $PASS FOR TYPEOUT
TYPDS ;; GO TYPE--DECIMAL ASCII WITH SIGN
TYPE $NULL ;; TYPE A NULL CHARACTER
$GET42: MOV #42,R0 ;; GET MONITOR ADDRESS
BEQ $DOAGN ;; BRANCH IF NO MONITOR
RESET ;; CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;; GO TO MONITOR
NOP ;; SAVE ROOM
NOP ;; FOR
NOP ;; ACT11
$DOAGN: JMP 2(PC)+ ;; RETURN
$RTNAD: .WORD PASSLP
$NULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
$SENDMG: .ASCIZ <15><12>/END PASS #/

HALT

;;*****
; INTERRUPT SUBROUTINE FOR CLOCK. BASIC CLOCK COUNT
; IS STORED IN CLKCT. LOCATION CLKCT+1 CONTAINS #OF
; ELAPSED SECONDS.
;;*****
TCLOCK: INCB CLKCT ;; INCREMENT CLOCK COUNT
CMPB CLKCT,#60. ;; BASIC COUNT = 1SECOND?
BNE 1$ ;; NO
CLRB CLKCT ;; YES RESET CLOCK ITERATION
INCB CLKCT+1 ;; AND INCREMENT SECOND COUNT
1$: BIC #BIT7,2LCLOCK ;; CLEAR FLAG.
RTI

;;*****
; SUBROUTINE TO CLEAR ERROR/DATA OUTPUT LOCATIONS. NEEDED
; ONLY WHEN DISPLAYING BYTES IN WORD LOCATIONS.
;;*****
CLREG: CLR $GDADR
CLR $BDADR
CLR $GDAT
CLR $BDAT

```

```

2501 011'64 000200          RIS  RO
2502
2503          ;;*****
2504          ;SUBROUTINE TO RUN THE 'SECOND' FLOPS VIA B007 IN
2505          ;MAINT MADE. R1= # OF ITERATIONS
2506          ;STARS 1
2507
2508 011466 052777 000004 170116 MSEC:  BIS  #MAINT ,@NCSR ;INSURE MAINT SET
2509 011474 052777 000020 170110      BIS  #BTENA,@NCSR ;ALLOW SECOND PROPOGATION.
2510 011502 052777 000200 170110 1$:  BIS  #SETSEC ,@NTIM
2511 011510 005777 170076      TST  @NCSR ;CLOCK THE SECOND COUNTER.
2512 011514 005301      DEC  R1 ;DECREMENT ITERATION COUNT
2513 011516 001371      BNE  1$ ;NOT DONE-LOOP
2514 011520 042777 000200 170072      BIC  #SETSEC ,@NTIM ;DISABLE THE MAINT/SERIAL CLOCK.
2515 011526 000200          RTS  RO
2516
2517          ;;*****
2518          ;SUBROUTINE TO GENERATE 4 COUNT LOW AND CLR 1 BY
2519          ;SUPPORTING THE T1-T2 SEQUENCE FOR A COUNT OF 5.
2520          ;;*****
2521
2522
2523 011530 005237 011616          FORCTA: INC  FOREXT ;SET UP TO WAIT UNTIL T2 ENDS.
2524 011534 032777 000400 170050 FORCT:  BIT  #TM1,@NCSR ;INSURE T-2 RESET
2525 011542 001374          BNE  .-6
2526 011544 012705 000005          MOV  #5 ,R5 ;SET UP A COUNT OF 5.
2527 011550 052777 000001 170034 1$:  BIS  #TGO,@NCSR ;START TM1 AND TM2
2528 011556 032777 000400 170026      BIT  #TM1 ,@NCSR
2529 011564 001374          BNE  .-6 ;WAIT FOR TM1 TO RESET
2530 011566 005305          DEC  R5 ;LOOP UNTIL COUNT = 0.
2531 011570 001367          BNE  1$
2532 011572 005737 011616          TST  FOREXT ;WAIT UNTIL T2 ENDS?
2533 011576 001406          BEQ  2$ ;NO
2534 011600 032777 001000 170004      BIT  #TM2,@NCSR ;YES
2535 011606 001374          BNE  .-6
2536 011610 005037 011616          CLR  FOREXT ;CLEAR FLAG BEFORE EXITING
2537 011614 000200          2$:  RTS  RO
2538 011616 000000          FOREXT: .WORD 0
2539
2540          ;;*****
2541          ;ROUTINE TO SYNC UP ON TIM3 WHICH IS A FREE-RUNNING
2542          ;1 SECOND CLOCK.
2543          ;;*****
2544
2545 011620 032777 000040 167764 SYNC3:  BIT  #TM3 ,@NCSR ;WAIT FOR TIME 3
2546 011626 001774          BEQ  .-6 ;TO GO HIGH
2547 011630 032777 000040 167754      BIT  #TM3 ,@NCSR
2548 011636 001374          BNE  .-6 ;AND EXIT WHEN IT GOES LOW.
2549 011640 000200          RTS  RO
2550 011642 032777 000040 167742 SYNC3A: BIT  #TM3,@NCSR ;WAIT FOR TIME 3NOT
2551 011650 001374          BNE  .-6
2552 011652 032777 000040 167732      BIT  #TM3,@NCSR ;EXIT ON TIME 3 TRUE
2553 011660 001774          BEQ  .-6
2554 011662 000200          RTS  RO
2555
2556

```

```

2557 ;*****
2558 ;SUBROUTINE TO LOOK FOR A CARRIAGE RETURN. IF FOUND
2559 ;EXIT IS ENTRY +2, OTHERWISE NORMAL EXIT
2560 ;*****
2561
2562 011664 032777 000200 167252 CKCRT: BIT #BIT7,2STKS ;RECEIVED A CHAR.?
2563 011672 001411 BEQ CRTXT ;NO-EXIT
2564 011674 117705 167246 MOVB 2STKB,R5 ;LOAD THE INPUT BUFFER
2565 011700 042705 000200 BIC #200,R5 ;CLEAR THE PARITY
2566 011704 123705 001165 CMPB $CRLF,R5 ;COMPARE FOR A CARRIAGE RET.
2567 011710 001002 BNE CRTXT ;NO-IGNORE IT
2568 011712 062700 000002 ADD #2,R0 ;YES-ADD 2 TO EXIT COUNT
2569 011716 000200 CRTXT: RTS R0 ;AND EXIT
2570
2571 ;*****
2572 ;SUBROUTINE TO START THE CLOCK.
2573 ;*****
2574
2575 011720 052777 000100 167704 GOCLK: BIS #100,2LCLOK ;ALLOW CLOCK INTERRUPTS.
2576 011726 042777 000200 167676 BIC #200,2LCLOK ;START THE CLOCK
2577 011734 005037 001644 CLR CLKCT ;CLEAR THE CLOCK COUNTER.
2578 011740 000200 RTS R0
2579
2580
2581 .SBTTL SCOPE HANDLER ROUTINE
2582
2583 ;*****
2584 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
2585 ;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
2586 ;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
2587 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2588 ;*SW14=1 LOOP ON TEST
2589 ;*SW11=1 INHIBIT ITERATIONS
2590 ;*SW09=1 LOOP ON ERROR
2591 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
2592 ;*CALL
2593 ;* SCOPE ;;SCOPE=IOT
2594
2595 $SCOPE:
2596 011742 032777 040000 167170 1$: BIT #BIT14,2SWR ;:LOOP ON PRESENT TEST?
2597 011750 001111 BNE $OVER ;:YES IF SW14=1
2598 ;*****START OF CODE FOR THE XOR TESTER*****
2599 011752 000416 $XTSTR: BR 6$ ;:IF RUNNING ON THE "XOR" TESTER CHANGE
2600 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
2601 011754 013746 000004 MOV 2$ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
2602 011760 012737 012000 000004 MOV 5$,2$ERRVEC ;:SET FOR TIMEOUT
2603 011766 005737 177060 TST 2$177060 ;:TIME OUT ON XOR?
2604 011772 012637 000004 MOV (SP)+,2$ERRVEC ;:RESTORE THE ERROR VECTOR
2605 011776 000463 BR $SVLAD ;:GO TO THE NEXT TEST
2606 012000 022626 5$: CMP (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
2607 012002 012637 000004 MOV (SP)+,2$ERRVEC ;:RESTORE THE ERROR VECTOR
2608 012006 000423 BR 7$ ;:LOOP ON THE PRESENT TEST
2609 012010 6$;*****END OF CODE FOR THE XOR TESTER*****
2610 012010 032777 000400 167122 BIT #BIT08,2SWR ;:LOOP ON SPEC. TEST?
2611 012016 001404 BEQ 2$ ;:BR IF NO
2612 012020 127737 167114 001102 CMPB 2$SWR,$STNM ;:ON THE RIGHT TEST? SWR<7:0>

```

```

2613 012026 001462          BEQ      $OVER          ;; BR IF YES
2614 012030 105737 001103   2$:  TSTB     $ERFLG          ;; HAS AN ERROR OCCURRED?
2615 012034 001421          BEQ      3$              ;; BR IF NO
2616 012036 123737 001115 001103  CMPB     $ERMAX,$ERFLG  ;; MAX. ERRORS FOR THIS TEST OCCURRED?
2617 012044 101015          BHI      3$              ;; BR IF NO
2618 012046 032777 001000 167064  BIT      %BIT09,%SWR    ;; LOOP ON ERROR?
2619 012054 001404          BEQ      4$              ;; BR IF NO
2620 012056 013737 001110 001106  7$:  MOV      $LPERR,$LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
2621 012064 000443          BR       $OVER          ;;
2622 012066 105037 001103   4$:  CLRB     $ERFLG          ;; ZERO THE ERROR FLAG
2623 012072 005037 001160          CLR      $TIMES        ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
2624 012076 000415          BR       1$              ;; ESCAPE TO THE NEXT TEST
2625 012100 032777 004000 167032  3$:  BIT      %BIT11,%SWR  ;; INHIBIT ITERATIONS?
2626 012106 001011          BNE     1$              ;; BR IF YES
2627 012110 005737 001100          TST     $PASS          ;; IF FIRST PASS OF PROGRAM
2628 012114 001406          BEQ     1$              ;; INHIBIT ITERATIONS
2629 012116 005237 001104          INC     $ICNT          ;; INCREMENT ITERATION COUNT
2630 012122 023737 001160 001104  CMP      $TIMES,$ICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
2631 012130 002021          BGE     $OVER          ;; BR IF MORE ITERATION REQUIRED
2632 012132 012737 000001 001104  1$:  MOV      #1,$ICNT     ;; REINITIALIZE THE ITERATION COUNTER
2633 012140 013737 012210 001160  MOV      $MXCNT,$TIMES ;; SET NUMBER OF ITERATIONS TO DO
2634 012146 105237 001102          SSVLAD: INCB     $STNM      ;; COUNT TEST NUMBERS
2635 012152 011637 001106          MOV     (SP),$LPADR    ;; SAVE SCOPE LOOP ADDRESS
2636 012156 011637 001110          MOV     (SP),$LPERR    ;; SAVE ERROR LOOP ADDRESS
2637 012162 005037 001162          CLR     $ESCAPE        ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
2638 012166 112737 000001 001115  MOVB    #1,$ERMAX      ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2639 012174 013777 001102 166740  $OVER: MOV     $STNM,%DISPLAY ;; DISPLAY TEST NUMBER
2640 012202 013716 001106          MOV     $LPADR,(SP)    ;; FUDGE RETURN ADDRESS
2641 012206 000002          RTI                    ;; FIXES PS
2642 012210 000010          $MXCNT: 10            ;; MAX. NUMBER OF ITERATIONS
2643          .SBTTL  ERROR HANDLER ROUTINE
2644
2645          ;; *****
2646          ;; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
2647          ;; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
2648          ;; *AND GO TO $ERRTYP ON ERROR
2649          ;; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2650          ;; *SW15=1      HALT ON ERROR
2651          ;; *SW13=1      INHIBIT ERROR TYPEOUTS
2652          ;; *SW09=1     LOOP ON ERROR
2653          ;; *CALL
2654          ;; *      ERROR  N      ;; ERROR=EMT AND N=ERROR ITEM NUMBER
2655
2656          $ERROR:
2657 012212 105237 001103   7$:  INCB     $ERFLG          ;; SET THE ERROR FLAG
2658 012216 001775          BEQ     7$              ;; DON'T LET THE FLAG GO TO ZERO
2659 012220 013777 001102 166714  MOV     $STNM,%DISPLAY  ;; DISPLAY TEST NUMBER AND ERROR FLAG
2660 012226 005237 001112          INC     $ERTTL        ;; INC THE ERROR COUNT
2661 012232 011637 001116          MOV     (SP),$ERRPC    ;; GET ADDRESS OF ERROR INSTRUCTION
2662 012236 162737 000002 001116  SUB     #2,$ERRPC       ;;
2663 012244 117737 166646 001114  MOVB   %$ERRPC,$ITEMB  ;; STRIP AND SAVE THE ERROR ITEM CODE
2664 012252 032777 020000 166660  BIT     %BIT13,%SWR    ;; SKIP TYPEOUT IF SET
2665 012260 001004          BNE     20$            ;; SKIP TYPEOUTS
2666 012262 004737 013204  JSR     PC,$ERRTYP     ;; GO TO USER ERROR ROUTINE
2667 012266 104400 001165          TYPE   ,SCLF
2668 012272

```

```

2669 012272 005777 166642      25:   TST      @SWR          ;; HALT ON ERROR
2670 012276 100001                BPL      3$           ;; SKIP IF CONTINUE
2671 012300 000000                HALT                    ;; HALT ON ERROR!
2672 012302 032777 001000 166630 35:   BIT      @BIT09,@SWR  ;; LOOP ON ERROR SWITCH SET?
2673 012310 001402                BEQ      4$           ;; BR IF NO
2674 012312 013716 001110                MOV      $LPERR,(SP)  ;; FUDGE RETURN FOR LOOPING
2675 012316 005737 001162      45:   TST      $ESCAPE     ;; CHECK FOR AN ESCAPE ADDRESS
2676 012322 001402                BEQ      5$           ;; BR IF NONE
2677 012324 013716 001162      55:   MOV      $ESCAPE,(SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE
2678 012330
2679 012330 022737 011352 000042      55:   CMP      @SENDAD,@#42 ;; ACT-11 AUTO-ACCEPT?
2680 012336 001001                BNE      6$           ;; BRANCH IF NO
2681 012340 000000                HALT                    ;; YES
2682 012342
2683 012342 000002      65:   RTI                    ;; RETURN
2684 .SBTTL  TTY INPUT ROUTINE
2685
2686 ;*****
2687 .ENABL  LSB
2688
2689 .DSABL  LSB
2690
2691
2692 ;*****
2693 ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2694 ;CALL:
2695 ;*   RDCHR          ;; INPUT A SINGLE CHARACTER FROM THE TTY
2696 ;*   RETURN HERE   ;; CHARACTER IS ON THE STACK
2697 ;*                ;; WITH PARITY BIT STRIPPED OFF
2698 ;*
2699 ;*
2700 $RDCHR: MOV      (SP),-(SP)  ;; PUSH DOWN THE PC
2701        MOV      4(SP),2(SP) ;; SAVE THE PS
2702 15:    TSTB     @STKS     ;; WAIT FOR
2703        BPL      1$       ;; A CHARACTER
2704        MOVB     @STKB,4(SP) ;; READ THE TTY
2705        BIC      @C<177>,4(SP) ;; GET RID OF JUNK IF ANY
2706        CMP      4(SP),#23  ;; IS IT A CONTROL-S?
2707        BNE      3$       ;; BRANCH IF NO
2708 25:    TSTB     @STKS     ;; WAIT FOR A CHARACTER
2709        BPL      2$       ;; LOOP UNTIL ITS THERE
2710        MOVB     @STKB,-(SP) ;; GET CHARACTER
2711        BIC      @C<177>,(SP) ;; MAKE IT 7-BIT ASCII
2712        CMP      (SP)+,#21  ;; IS IT A CONTROL-Q?
2713        BNE      2$       ;; IF NOT DISCARD IT
2714        BR       1$       ;; YES, RESUME
2715 35:    CMP      4(SP),#140  ;; IS IT UPPER CASE?
2716        BLT      4$       ;; BRANCH IF YES
2717        CMP      4(SP),#175  ;; IS IT A SPECIAL CHAR?
2718        BGT      4$       ;; BRANCH IF YES
2719        BIC      @40,4(SP)  ;; MAKE IT UPPER CASE
2720 45:    RTI                    ;; GO BACK TO USER
2721 ;*****
2722 ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2723 ;CALL:
2724 ;*   RDLIN          ;; INPUT A STRING FROM THE TTY

```

```

2725          : *      RETURN HERE          :: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2726          : *                                :: TERMINATOR WILL BE A BYTE OF ALL 0'S
2727
2728 012464 010346 SRDLIN: MOV      R3, -(SP)          :: SAVE R3
2729 012466 005046 CLR      -(SP)          :: CLEAR THE RUBOUT KEY
2730 012470 012703 012720 1S:  MOV      #STTYIN, R3      :: GET ADDRESS
2731 012474 022703 012730 2S:  CMP      #STTYIN+8., R3      :: BUFFER FULL?
2732 012500 101456 BLOS    4S          :: BR IF YES
2733 012502 104405 R0CHR   :: GO READ ONE CHARACTER FROM THE TTY
2734 012504 112613 MOVB    (SP)+, (R3)    :: GET CHARACTER
2735 012506 122713 000177 10S: CMPB    #177, (R3)    :: IS IT A RUBOUT
2736 012512 001022 BNE     5S          :: BR IF NO
2737 012514 005716 TST     (SP)          :: IS THIS THE FIRST RUBOUT?
2738 012516 001007 BNE     6S          :: BR IF NO
2739 012520 112737 000134 012716 MOVB    #' \, 9S     :: TYPE A BACK SLASH
2740 012526 104400 012716 TYPE     9S
2741 012532 012716 177777 MOV     1-1, (SP)    :: SET THE RUBOUT KEY
2742 012536 005303 6S:  DEC     R3          :: BACKUP BY ONE
2743 012540 020327 012720 CMP     R3, #STTYIN  :: STACK EMPTY?
2744 012544 103434 BLO     4S          :: BR IF YES
2745 012546 111337 012716 MOVB    (R3), 9S     :: SETUP TO TYPEOUT THE DELETED CHAR.
2746 012552 104400 012716 TYPE     9S          :: GO TYPE
2747 012556 000746 BR      2S          :: GO READ ANOTHER CHAR.
2748 012560 005716 5S:  TST     (SP)          :: RUBOUT KEY SET?
2749 012562 001406 BEQ     7S          :: BR IF NO
2750 012564 112737 000134 012716 MOVB    #' \, 9S     :: TYPE A BACK SLASH
2751 012572 104400 012716 TYPE     9S
2752 012576 005016 CLR     (SP)          :: CLEAR THE RUBOUT KEY
2753 012600 122713 000025 7S:  CMPB    #25, (R3)    :: IS CHARACTER A CTRL U?
2754 012604 001003 BNE     8S          :: BR IF NO
2755 012606 104400 012730 TYPE     , SCNTLU    :: TYPE A CONTROL "U"
2756 012612 000726 BR      1S          :: GO START OVER
2757 012614 122713 000022 8S:  CMPB    #22, (R3)    :: IS CHARACTER A "↑R"?
2758 012620 001011 BNE     3S          :: BRANCH IF NO
2759 012622 105013 CLRB   (R3)          :: CLEAR THE CHARACTER
2760 012624 104400 001165 TYPE     , SCRLF    :: TYPE A "CR" & "LF"
2761 012630 104400 012720 TYPE     , STTYIN    :: TYPE THE INPUT STRING
2762 012634 000717 BR      2S          :: GO PICKUP ANOTHER CHARACTER
2763 012636 104400 001164 4S:  TYPE     , SQUES    :: TYPE A '?'
2764 012642 000712 BR      1S          :: CLEAR THE BUFFER AND LOOP
2765 012644 111337 012716 3S:  MOVB    (R3), 9S     :: ECHO THE CHARACTER
2766 012650 104400 012716 TYPE     9S
2767 012654 122723 000015 CMPB    #15, (R3)+  :: CHECK FOR RETURN
2768 012660 001305 BNE     2S          :: LOOP IF NOT RETURN
2769 012662 105063 177777 CLRB   -1(R3)       :: CLEAR RETURN (THE 15)
2770 012666 104400 001166 TYPE     , LF        :: TYPE A LINE FEED
2771 012672 005726 TST     (SP)+       :: CLEAN RUBOUT KEY FROM THE STACK
2772 012674 012603 MOV     (SP)+, R3    :: RESTORE R3
2773 C:2676 011646 MOV     (SP), -(SP)  :: ADJUST THE STACK AND PUT ADDRESS OF THE
2774 012700 016666 000004 000002 MOV     4(SP), 2(SP) :: FIRST ASCII CHARACTER ON IT
2775 012706 012766 012720 000004 MOV     #STTYIN, 4(SP)
2776 012714 000002 RTI
2777 012716 000 9S:  .BYTE 0          :: RETURN
2778 012717 000 .BYTE 0          :: STORAGE FOR ASCII CHAR. TO TYPE
2779 012720 000010 .BLKB 8.          :: TERMINATOR
2780 012730 052536 005015 000 $TTYIN: .BLKB 8.          :: RESERVE 8 BYTES FOR TTY INPUT
SCNTLU: .ASCIZ /↑U<15><12> :: CONTROL "U"

```

```

2781 012735 136 006507 000012 $CNTLG: .ASCIZ /IG<15><12> ; ;CONTROL "G"
2782 012742 005015 053523 020122 $MSWR: .ASCIZ <15><12>/SWR = /
2783 012750 020075 000
2784 012753 040 047040 053505 $MNEW: .ASCIZ / NEW = /
2785 012760 036440 0C0040
2786 .SBTTL TYPE ROUTINE
2787
2788 *****
2789 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2790 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2791 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2792 *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2793 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2794 *
2795 *CALL:
2796 *1) USING A TRAP INSTRUCTION
2797 * TYPE ,MESADR ; ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2798 *OR
2799 * TYPE
2800 * MESADR
2801 *
2802
2803 012764 105737 001157 $TYPE: TSTB $TFPLG ; ; IS THERE A TERMINAL?
2804 012770 100002 BPL IS ; ; BR IF YES
2805 012772 000000 HALT ; ; HALT HERE IF NO TERMINAL
2806 012774 000407 BR 3$ ; ; LEAVE
2807 012776 010046 1$: MOV RO, -(5$) ; ; SAVE RO
2808 013000 017600 000002 MOV 22(SP), RO ; ; GET ADDRESS OF ASCIZ STRING
2809 013004 112046 2$: MOVB (RO)+, -(5$) ; ; PUSH CHARACTER TO BE TYPED ONTO STACK
2810 013006 001005 BNE 4$ ; ; BR IF IT ISN'T THE TERMINATOR
2811 013010 005726 TST (5$)+ ; ; IF TERMINATOR POP IT OFF THE STACK
2812 013012 012600 60$: MOV (5$)+, RO ; ; RESTORE RO
2813 013014 062716 3$: ADD 2, (5$) ; ; ADJUST RETURN PC
2814 013020 000002 RTI ; ; RETURN
2815 013022 122716 4$: CMPB #HT, (5$) ; ; BRANCH IF <HT>
2816 013026 001430 BEQ 8$ ; ;
2817 013030 122716 000200 CMPB #CRLF, (5$) ; ; BRANCH IF NOT <CRLF>
2818 013034 001006 BNE 5$ ; ;
2819 013036 005726 TST (5$)+ ; ; POP <CR><LF> EQUIV
2820 013040 104400 TYPE ; ; TYPE A CR AND LF
2821 013042 001165 $CRLF
2822 013044 105037 013200 CLRB $CHARCNT ; ; CLEAR CHARACTER COUNT
2823 013050 000755 BR 2$ ; ; GET NEXT CHARACTER
2824 013052 004737 013134 5$: JSR PC, $TYPEC ; ; GO TYPE THIS CHARACTER
2825 013056 123726 001156 6$: CMPB $FILLC, (5$)+ ; ; IS IT TIME FOR FILLER CHARS.?
2826 013062 001350 BNE 2$ ; ; IF NO GO GET NEXT CHAR.
2827 013064 013746 001154 MOV $NULL, -(5$) ; ; GET # OF FILLER CHARS. NEEDED
2828 ; ; AND THE NULL CHAR.
2829 013070 105366 000001 7$: DECB 1(5$) ; ; DOES A NULL NEED TO BE TYPED?
2830 013074 002770 BLT 6$ ; ; BR IF NO--GO POP THE NULL OFF OF STACK
2831 013076 004737 013134 JSR PC, $TYPEC ; ; GO TYPE A NULL
2832 013102 105337 013200 DECB $CHARCNT ; ; DO NOT COUNT AS A COUNT
2833 013106 000770 BR 7$ ; ; LOOP
2834
2835 ;HORIZONTAL TAB PROCESSOR
2836

```

```

2837 013110 112716 000040 8$: MOV B 0' (SP)      ;; REPLACE TAB WITH SPACE
2838 013114 00737 013134 9$: JSR PC,$TYPEC    ;; TYPE A SPACE
2839 013120 132737 000007 013200 BITB 07,$CHARCNT  ;; BRANCH IF NOT AT
2840 013126 00137? BNE 9$          ;; TAB STOP
2841 013120 005726 TST (SP)+      ;; POP SPACE OFF STACK
2842 013132 000724 BR 2$          ;; GET NEXT CHARACTER
2843 013134 105777 166010 $TYPEC: TSTB 0$TPS    ;; WAIT UNTIL PRINTER IS READY
2844 013140 100375 BPL $TYPEC
2845 013142 116677 100002 166002 MOVE 2(SP),0$TPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.
2846 013150 122766 000015 000002 CMPS 0CR,2(SP)  ;; IS CHARACTER A CARRIAGE RETURN?
2847 013156 001003 BNE 1$          ;; BRANCH IF NO
2848 013160 105037 013200 CLRB $CHARCNT  ;; YES--CLEAR CHARACTER COUNT
2849 013164 000406 BR $TYPEX      ;; EXIT
2850 013166 122766 000012 000002 1$: CMPB 0LF,2(SP)  ;; IS CHARACTER A LINE FEED?
2851 013174 001402 BEQ $TYPEX      ;; BRANCH IF YES
2852 013176 105227 INCB (PC)+      ;; COUNT THE CHARACTER
2853 013200 000000 $CHARCNT: .WORD 0  ;; CHARACTER COUNT STORAGE
2854 013202 000207 $TYPEX: RTS PC

```

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

*****
; THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
; ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
; AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

```

```

$ERRTYP:
TYPE $SCLF      ;; "CARRIAGE RETURN" & "LINE FEED"
MOV R0,-(SP)   ;; SAVE R0
CLR R0         ;; PICKUP THE ITEM INDEX
BISB 00,$ITEMB,R0
BNE 1$        ;; IF ITEM NUMBER IS ZERO, JUST
              ;; TYPE THE PC OF THE ERROR
MOV $ERRPC,-(SP)  ;; SAVE $ERRPC FOR TYPEOUT
              ;; ERROR ADDRESS
              ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
              ;; GET OUT
1$: DEC R0      ;; ADJUST THE INDEX SO THAT IT WILL
      ASL R0   ;; WORK FOR THE ERROR TABLE
      ASL R0
      ASL R0
      ADD 0,$ERRTB,R0  ;; FORM TABLE POINTER
      MOV (R0)+,2$    ;; PICKUP "ERROR MESSAGE" POINTER
      BEQ 3$         ;; SKIP TYPEOUT IF NO POINTER
      TYPE          ;; TYPE THE "ERROR MESSAGE"
2$: .WORD 0        ;; "ERROR MESSAGE" POINTER GOES HERE
      TYPE $SCLF  ;; "CARRIAGE RETURN" & "LINE FEED"
3$: MOV (R0)+,4$  ;; PICKUP "DATA HEADER" POINTER
      BEQ 5$      ;; SKIP TYPEOUT IF 0
      TYPE        ;; TYPE THE "DATA HEADER"
4$: .WORD 0        ;; "DATA HEADER" POINTER GOES HERE
      TYPE $SCLF  ;; "CARRIAGE RETURN" & "LINE FEED"
5$: MOV R1,-(SP)  ;; SAVE R1
      MOV (R0)+,R1  ;; PICKUP "DATA TABLE" POINTER
      BEQ 9$      ;; BR IF NO DATA TO BE TYPED
      MOV (R0)+,R0  ;; PICKUP "DATA FORMAT" POINTER

```

```

2863 013204 104400 001165
2864 013204 104400 001165
2865 013210 010046
2866 013212 005000
2867 013214 153700 001114
2868 013220 001004
2869
2870 013222 013746 001116
2871
2872 013226 104401
2873 013230 000445
2874 013232 005300 1$: DEC R0
2875 013234 006300 ASL R0
2876 013236 006300 ASL R0
2877 013240 006300 ASL R0
2878 013242 062700 001170 ADD 0,$ERRTB,R0
2879 013246 012037 013256 MOV (R0)+,2$
2880 013252 001404 BEQ 3$
2881 013254 104400 TYPE
2882 013256 000000 2$: .WORD 0
2883 013260 104400 001165 TYPE $SCLF
2884 013264 012037 013274 3$: MOV (R0)+,4$
2885 013270 001404 BEQ 5$
2886 013272 104400 TYPE
2887 013274 000000 4$: .WORD 0
2888 013276 104400 001165 TYPE $SCLF
2889 013302 010146 5$: MOV R1,-(SP)
2890 013304 012001 MOV (R0)+,R1
2891 013306 001415 BEQ 9$
2892 013310 012000 MOV (R0)+,R0

```

E05

```

2893 013312 105720      6S:   TSTB   (R0)+      ;; "OCTAL" OR "DECIMAL"
2894 013314 001003      ENE    7S          ;; BR IF DECIMAL
2895 013316 013146      MOV    2(R1)+,-(SP) ;; SAVE 2(R1)+ FOR TYPEOUT
2896 013320 104401      TYPOC                ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
2897 013322 000402      BR     8S
2898 013324                7S:
2899 013324 013146      MOV    2(R1)+,-(SP) ;; SAVE 2(R1)+ FOR TYPEOUT
2900 013326 104404      TYPDS                ;; GO TYPE--DECIMAL ASCII WITH SIGN
2901 013330 005711      8S:   TST    (R1)       ;; IS THERE ANOTHER NUMBER?
2902 013332 001403      BEQ    9S          ;; BR IF NO
2903 013334 104400 0..3354  TYPE   11S        ;; TYPE TWO(2) SPACES
2904 013340 000764      BR     6S          ;; LOOP
2905
2906 013342 012601      9S:   MOV    (SP)+,R1   ;; RESTORE R1
2907 013344 012600      10S:  MOV    (SP)+,R0   ;; RESTORE R0
2908 013346 104400 001155  TYPE   $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
2909 013352 000207      RTS    PC         ;; RETURN
2910 013354 020040 000      11S:  .ASCIZ / /       ;; TWO(2) SPACES
2911      .EVEN
2912      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
2913
2914      ;; *****
2915      ;; THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2916      ;; OCTAL (ASCII) NUMBER AND TYPE IT.
2917      ;; $TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2918      ;; CALL:
2919      ;;     MOV     NUM,-(SP)      ;; NUMBER TO BE TYPED
2920      ;;     TYPOS                ;; CALL FOR TYPEOUT
2921      ;;     .BYTE  N              ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2922      ;;     .BYTE  M              ;; M=1 OR 0
2923      ;;                               ;; 1=TYPE LEADING ZEROS
2924      ;;                               ;; 0=SUPPRESS LEADING ZEROS
2925
2926      ;; $TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2927      ;; $TYPOS OR $TYPOC
2928      ;; CALL:
2929      ;;     MOV     NUM,-(SP)      ;; NUMBER TO BE TYPED
2930      ;;     TYPON                ;; CALL FOR TYPEOUT
2931
2932      ;; $TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2933      ;; CALL:
2934      ;;     MOV     NUM,-(SP)      ;; NUMBER TO BE TYPED
2935      ;;     TYPOC                ;; CALL FOR TYPEOUT
2936
2937 013360 017646 000000      STYPOS: MOV    2(SP),-(SP) ;; PICKUP THE MODE
2938 013364 116637 000001 013603  MOVB   1(SP),SOFILL ;; LOAD ZERO FILL SWITCH
2939 013372 112637 013605      MOVB   (SP)+,$OMODE+1 ;; NUMBER OF DIGITS TO TYPE
2940 013376 062716 000002      ADD    #2,(SP)      ;; ADJUST RETURN ADDRESS
2941 013402 000406      BR     $TYPON
2942 013404 112737 000001 013603  STYPOC: MOVB   #1,$SOFILL ;; SET THE ZERO FILL SWITCH
2943 013412 112737 000006 013605      MOVB   #6,$OMODE+1 ;; SET FOR SIX(6) DIGITS
2944 013420 112737 000005 013602  STYPON: MOVB   #5,$OCNT  ;; SET THE ITERATION COUNT
2945 013426 010346      MOV    R3,-(SP)    ;; SAVE R3
2946 013430 010446      MOV    R4,-(SP)    ;; SAVE R4
2947 013432 010546      MOV    R5,-(SP)    ;; SAVE R5
2948 013434 113704 013605      MOVB   $OMODE+1,R4 ;; GET THE NUMBER OF DIGITS TO TYPE

```

```

2949 013440 005404          NEG      R4
2950 013442 062704 000006  ADD      #6,R4          ;; SUBTRACT IT FOR MAX. ALLOWED
2951 013446 110437 013604  MOVVB   R4,$OMODE     ;; SAVE IT FOR USE
2952 013452 113704 013603  MOVVB   $OFILL,R4    ;; GET THE ZERO FILL SWITCH
2953 013456 016605 0C0012  MOV     12(SP),R5    ;; PICKUP THE INPUT NUMBER
2954 013462 005003          CLR      R3          ;; CLEAR THE OUTPUT WORD
2955 013464 006105          15:    ROL      R5          ;; ROTATE MSB INTO "C"
2956 013466 000404          BR      35          ;; GO DO MSB
2957 013470 006105          25:    ROL      R5          ;; FORM THIS DIGIT
2958 013472 006105          ROL      R5
2959 013474 006105          ROL      R5
2960 013476 010503          MOV     R5,R3
2961 013500 006103          35:    ROL      R3          ;; GET LSB OF THIS DIGIT
2962 013502 105337 013604  DECB   $OMODE        ;; TYPE THIS DIGIT?
2963 013506 100016          BPL     75          ;; BR IF NO
2964 013510 042703 177770  BIC     #177770,R3   ;; GET RID OF JUNK
2965 013514 001002          BNE     45          ;; TEST FOR 0
2966 013516 005704          TST     R4          ;; SUPPRESS THIS 0?
2967 013520 001403          BEQ     55          ;; BR IF YES
2968 013522 005204          45:    INC      R4          ;; DON'T SUPPRESS ANYMORE 0'S
2969 013524 052703 000060  BIS     #'0,R3      ;; MAKE THIS DIGIT ASCII
2970 013530 052703 000040  55:    BIS     #' ,R3   ;; MAKE ASCII IF NOT ALREADY
2971 013534 110337 013600  MOVVB   R3,$S          ;; SAVE FOR TYPING
2972 013540 104400 013600  TYPE    8$          ;; GO TYPE THIS DIGIT
2973 013544 105337 013602  75:    DECB   $OCNT    ;; COUNT BY 1
2974 013550 003347          BGT     25          ;; BR IF MORE TO DO
2975 013552 0C 102          BLT     65          ;; BR IF DONE
2976 013554 005204          INC     R4          ;; INSURE LAST DIGIT ISN'T A BLANK
2977 013556 000744          BR      25          ;; GO DO THE LAST DIGIT
2978 013560 012605          65:    MOV     (SP)+,R5   ;; RESTORE R5
2979 013562 012604          MOV     (SP)+,R4   ;; RESTORE R4
2980 013564 012603          MOV     (SP)+,R3   ;; RESTORE R3
2981 013566 016666 000002 0C0004  MOV     2(SP),4(SP) ;; SET THE STACK FOR RETURNING
2982 013574 012616          MOV     (SP)+,(SP)
2983 013576 000002          RTI          ;; RETURN
2984 013600          85:    .BYTE   0          ;; STORAGE FOR ASCII DIGIT
2985 013601          .BYTE   0          ;; TERMINATOR FOR TYPE ROUTINE
2986 013602          .BYTE   0          ;; OCTAL DIGIT COUNTER
2987 013603          .BYTE   0          ;; ZERO FILL SWITCH
2988 013604 000001  .WORD   0          ;; NUMBER OF DIGITS TO TYPE
2989          .SBTTL READ AN OCTAL NUMBER FROM THE TTY
2990
2991          ;; *****
2992          ;; THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
2993          ;; CHANGE IT TO BINARY.
2994          ;; CALL:
2995          ;;
2996          ;; RDOCT          ;; READ AN OCTAL NUMBER
2997          ;; RETURN HERE  ;; LOW ORDER BITS ARE ON TOP OF THE STACK
2998          ;;          ;; HIGH ORDER BITS ARE IN $HI OCT
2999 013606 011646          $RDOCT: MOV     (SP)-,(SP) ;; PROVIDE SPACE FOR THE
3000 013610 016666 000004 000002  MOV     4(SP),2(SP) ;; INPUT NUMBER
3001 013616 010046          MOV     R0,-(SP)   ;; PUSH R0 ON STACK
3002 013620 010146          MOV     R1,-(SP)   ;; PUSH R1 ON STACK
3003 013622 010246          MOV     R2,-(SP)   ;; PUSH R2 ON STACK
3004 013624 104406          15:    RDLIN          ;; READ AN ASCII LINE

```

```

3005 013626 012600      MOV      (SP)+,R0      ;; GET ADDRESS OF 1ST CHARACTER
3006 013630 005001      CLR      R1           ;; CLEAR DATA WORD
3007 013632 005002      CLR      R2
3008 013634 112046      2$:     MOVVB     (R0)+,-(SP)  ;; PICKUP THIS CHARACTER
3009 013636 001412      BEQ      3$          ;; IF ZERO GET OUT
3010 013640 006301      ASL      R1           ;; *2
3011 013642 006102      ROL      R2
3012 013644 006301      ASL      R1           ;; *4
3013 013646 006102      ROL      R2
3014 013650 006301      ASL      R1           ;; *8
3015 013652 006102      ROL      R2
3016 013654 042716 1:7770 BIC      #1C7,(SP)    ;; STRIP THE ASCII JUNK
3017 013660 062601      ADD      (SP)+,R1    ;; ADD IN THIS DIGIT
3018 013662 000764      BR       2$          ;; LOOP
3019 013664 005726      3$:     TST      (SP)+    ;; CLEAN TERMINATOR FROM STACK
3020 013666 010166 000012 MOV      R1,12(SP)   ;; SAVE THE RESULT
3021 013672 010237 01:706 MOV      R2,$SHIOCT
3022 013676 012602      MOV      (SP)+,R2    ;; POP STACK INTO R2
3023 013700 012601      MOV      (SP)+,R1    ;; POP STACK INTO R1
3024 013702 012600      MOV      (SP)+,R0    ;; POP STACK INTO R0
3025 013704 000002      RTI
3026 013706 000000      $SHIOCT: .WORD      0      ;; HIGH ORDER BITS GO HERE
3027
3028      .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3029
3030      ;*****
3031      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
3032      ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
3033      ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
3034      ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
3035      ;REPLACED WITH SPACES.
3036      ;CALL:
3037      ;*   MOV      NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
3038      ;*   TYPDS      ;; GO TO THE ROUTINE
3039
3040      $TYPDS:
3041      MOV      R0,-(SP)      ;; PUSH R0 ON STACK
3042      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
3043      MOV      R2,-(SP)      ;; PUSH R2 ON STACK
3044      MOV      R3,-(SP)      ;; PUSH R3 ON STACK
3045      MOV      R5,-(SP)      ;; PUSH R5 ON STACK
3046      MOV      #20200,-(SP)  ;; SET BLANK SWITCH AND SIGN
3047      MOV      20(SP),R5    ;; GET THE INPUT NUMBER
3048      BPL      1$          ;; BR IF INPUT IS POS.
3049      NEG      R5           ;; MAKE THE BINARY NUMBER POS.
3050      MOVVB     #'-,1(SP)    ;; MAKE THE ASCII NUMBER NEG.
3051      CLR      R0           ;; ZERO THE CONSTANT'S INDEX
3052      MOV      #50BLK,R3    ;; SETUP THE OUTPUT POINTER
3053      MOVVB     #'',(R3)+    ;; SET THE FIRST CHARACTER TO A BLANK
3054      CLR      R2           ;; CLEAR THE BCD NUMBER
3055      MOV      $DTBL(R0),R1  ;; GET THE CONSTANT
3056      3$:     SUB      R1,R5    ;; FORM THIS BCD DIGIT
3057      BLT      4$          ;; BR IF DONE
3058      INC      R2           ;; INCREASE THE BCD DIGIT BY 1
3059      BR       3$
3060      4$:     ADD      R1,R5    ;; ADD BACK THE CONSTANT
    
```

H05

MAINDEC-11-DZNA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 60
DZNA.SRC CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```
3061 013776 005702          TST      R2          ;; CHECK IF BCD DIGIT=0
3062 014000 001002          BNE     5$          ;; FALL THROUGH IF 0
3063 014002 105716          TSTB   (SP)        ;; STILL DOING LEADING 0'S?
3064 014004 100407          BMI     7$          ;; BR IF YES
3065 014006 106316          ASLB   (SP)        ;; MSD?
3066 014010 103003          BCC     6$          ;; BR IF NO
3067 014012 116663 000001 177777  MOVB   1(SP),-1(R3) ;; YES--SET THE SIGN
3068 014020 052702 000060 6$:     BIS   #'0,R2    ;; MAKE THE BCD DIGIT ASCII
3069 014024 052702 000040 7$:     BIS   #' ,R2    ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
3070 014030 110223          MOVB   R2,R3)+    ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
3071 014032 005720          TST   (R0)+      ;; JUST INCREMENTING
3072 014034 020027 000010  CMP   R0,#10     ;; CHECK THE TABLE INDEX
3073 014040 002746          BLT   2$          ;; GO DO THE NEXT DIGIT
3074 014042 003002          BGT   8$          ;; GO TO EXIT
3075 014044 010502          MOV   R5,R2      ;; GET THE LSD
3076 014046 000764          BR    6$          ;; GO CHANGE TO ASCII
3077 014050 105726          TSTB  (SP)+      ;; WAS THE LSD THE FIRST NON-ZERO?
3078 014052 100003          BPL   9$          ;; BR IF NO
3079 014054 116663 177777 177776  MOVB  -1(SP),-2(R3) ;; YES--SET THE SIGN FOR TYPING
3080 014062 105013          CLRB  (R3)       ;; SET THE TERMINATOR
3081 014064 012605          MOV   (SP)+,R5   ;; POP STACK INTO R5
3082 014066 012603          MOV   (SP)+,R3   ;; POP STACK INTO R3
3083 014070 012602          MOV   (SP)+,R2   ;; POP STACK INTO R2
3084 014072 012601          MOV   (SP)+,R1   ;; POP STACK INTO R1
3085 014074 012600          MOV   (SP)+,R0   ;; POP STACK INTO R0
3086 014076 104400 014124  TYPE  $DBLK      ;; NOW TYPE THE NUMBER
3087 014102 016666 000002 000004  MOV   2(SP),4(SP) ;; ADJUST THE STACK
3088 014110 012616          MOV   (SP)+,(SP)
3089 014112 000002          RTI                    ;; RETURN TO USER
3090 014114 023420          SOTBL: 1000.
3091 014116 001750          1000.
3092 014120 000144          100.
3093 014122 000012          10.
3094 014124 000004          $DBLK: .BLKW 4
3095          .SBTTL POWER DOWN AND UP ROUTINES
3096
3097          ;; *****
3098          : POWER DOWN ROUTINE
3099 014134 012737 014274 000024 $PWRDN: MOV   $SILLUP,2$PWRVEC ;; SET FOR FAST UP
3100 014142 012737 000340 000026  MOV   #340,2$PWRVEC+2 ;; PRIO:7
3101 014150 010046          MOV   R0,-(SP)    ;; PUSH R0 ON STACK
3102 014152 010146          MOV   R1,-(SP)    ;; PUSH R1 ON STACK
3103 014154 010246          MOV   R2,-(SP)    ;; PUSH R2 ON STACK
3104 014156 010346          MOV   R3,-(SP)    ;; PUSH R3 ON STACK
3105 014160 010446          MOV   R4,-(SP)    ;; PUSH R4 ON STACK
3106 014162 010546          MOV   R5,-(SP)    ;; PUSH R5 ON STACK
3107 014164 017746 164750  MOV   2$SR,-(SP)   ;; PUSH 2$SR ON STACK
3108 014170 010637 014206  MOV   SP,$SAVR6    ;; SAVE SP
3109 014174 012737 014206 000024  MOV   $PWRUP,2$PWRVEC ;; SET UP VECTOR
3110 014202 000000          HALT
3111 014204 000776          BR    .-2        ;; HANG UP
3112
3113          ;; *****
3114          : POWER UP ROUTINE
3115 014206 012737 014274 000024 $PWRUP: MOV   $SILLUP,2$PWRVEC ;; SET FOR FAST DOWN
3116 014214 013706 014300          MOV   $SAVR6,SP   ;; GET SP
```

```

3117 014220 005037 014300          CLR    $SAVR5          ;; WAIT LOOP FOR THE TTY
3118 014224 005237 014300          1$: INC    $SAVR6          ;; WAIT FOR THE INC
3119 014230 001375                   BNE    1$              ;; OF WORD
3120 014232 012677 164702          MOV    (SP)+, @SWR     ;; POP STACK INTO @SWR
3121 014236 012605                   MOV    (SP)+, R5      ;; POP STACK INTO R5
3122 014240 012604                   MOV    (SP)+, R4      ;; POP STACK INTO R4
3123 014242 012603                   MOV    (SP)+, R3      ;; POP STACK INTO R3
3124 014244 012602                   MOV    (SP)+, R2      ;; POP STACK INTO R2
3125 014246 012601                   MOV    (SP)+, R1      ;; POP STACK INTO R1
3126 014250 012600                   MOV    (SP)+, R0      ;; POP STACK INTO R0
3127 014252 012737 014134 000024    MOV    #SPWRDN, @PWRVEC ;; SET UP THE POWER DOWN VECTOR
3128 014260 012737 000340 000026    MOV    #340, @PWRVEC+2 ;; PRIO:7
3129 014266 104400                   TYPE   ;; REPORT THE POWER FAILURE
3130 014270 014302          $PWRMG: .WORD    $POWER ;; POWER FAIL MESSAGE POINTER
3131 014272 000002                   RTI                               ;;
3132 014274 000000          $ILLUP: HALT          ;; THE POWER UP SEQUENCE WAS STARTED
3133 014276 000776                   BR     .-2                    ;; BEFORE THE POWER DOWN WAS COMPLETE
3134 014300 000000          $SAVR6: 0                    ;; PUT THE SP HERE
3135 014302 005015 047520 042527    $POWER: .ASCIZ  <15><12>"POWER"
3136 014310 000122
3137
3138          .SBTTL  .EVEN
3139          .SBTTL  TRAP DECODER
3140
3141          ;; *****
3142          ;; *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3143          ;; *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3144          ;; *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3145          ;; *GO TO THAT ROUTINE.
3146 014312 010046          $TRAP: MOV    RO, -(SP)          ;; SAVE RO
3147 014314 016600 000002          MOV    2(SP), RO        ;; GET TRAP ADDRESS
3148 014320 005740          TST    -(RO)            ;; BACKUP BY 2
3149 014322 111000          MOV    (RO), RO         ;; GET RIGHT BYTE OF TRAP
3150 014324 006300          ASL    RO                ;; POSITION FOR INDEXING
3151 014326 016000 014334          MOV    $TRPAD(RO), RO   ;; INDEX TO TABLE
3152 014332 000200          RTS    RO                ;; GO TO ROUTINE
3153
3154          .SBTTL  TRAP TABLE
3155
3156          ;; *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3157          ;; *BY THE "TRAP" INSTRUCTION.
3158
3159          ;          ROUTINE
3160          ;          -----
3161 014334          $TRPAD:
3162 014334 012764          $TYPE  ;; CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
3163 014336 013404          $TYPOC ;; CALL=TYPOC     TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3164 014340 013360          $TYPOS ;; CALL=TYPOS     TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3165 014342 013420          $TYPON ;; CALL=TYPON      TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3166 014344 013710          $TYPDS ;; CALL=TYPDS     TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
3167
3168
3169 014346 012344          $RDCHR ;; CALL=RDCHR    TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
3170 014350 012464          $RDLIN ;; CALL=RDLIN    TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
3171 014352 013606          $RDOCT ;; CALL=RDOCT     TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
3172
    
```

| | | | | | |
|------|--------|--------|--------|--------|---|
| 3173 | | | | | |
| 3174 | | | | | |
| 3175 | | | | | |
| 3176 | 014354 | 053516 | 030461 | 052040 | DSETUP: .ASCII /NW11 TEST. PLACE UNIT IN FOLLOWING STATE:/(15)<12> |
| 3177 | 014362 | 051505 | 027124 | 050040 | |
| 3178 | 014370 | 040514 | 042503 | 052440 | |
| 3179 | 014376 | 044516 | 020124 | 047111 | |
| 3180 | 014404 | 043040 | 046117 | 047514 | |
| 3181 | 014412 | 044527 | 043516 | 051440 | |
| 3182 | 014420 | 040524 | 042524 | 006472 | |
| 3183 | 014426 | 012 | | | |
| 3184 | 014427 | 120 | 031055 | 041450 | .ASCII /P-2(CUST. CONN.)UNPLUGGED,MAINT. SWITCH TO ON, ALL PANEL SWITCHES DOWN/ |
| 3185 | 014434 | 051525 | 027124 | 041440 | |
| 3186 | 014442 | 047117 | 027116 | 052451 | |
| 3187 | 014450 | 050116 | 052514 | 043507 | |
| 3188 | 014456 | 042105 | 046454 | 044501 | |
| 3189 | 014464 | 052116 | 020056 | 053523 | |
| 3190 | 014472 | 052111 | 044103 | 052040 | |
| 3191 | 014500 | 020117 | 047117 | 020054 | |
| 3192 | 014506 | 046101 | 020114 | 040520 | |
| 3193 | 014514 | 042516 | 020114 | 053523 | |
| 3194 | 014522 | 052111 | 044103 | 051505 | |
| 3195 | 014530 | 042040 | 053517 | 006516 | |
| 3196 | 014536 | 012 | | | |
| 3197 | 014537 | 042 | 047105 | 042524 | .ASCII /"ENTER SWITCHLOCK" TO "DISABLE" AND "ALARM-LJAD" SWITCH TO "DISABLE"/<1 |
| 3198 | 014544 | 020122 | 053523 | 052111 | |
| 3199 | 014552 | 044103 | 047514 | 045503 | |
| 3200 | 014560 | 020042 | 047524 | 021040 | |
| 3201 | 014566 | 044504 | 040523 | 046102 | |
| 3202 | 014574 | 021105 | 040440 | 042116 | |
| 3203 | 014602 | 021040 | 046101 | 051101 | |
| 3204 | 014610 | 026515 | 047514 | 042101 | |
| 3205 | 014616 | 020042 | 053523 | 052111 | |
| 3206 | 014624 | 044103 | 052040 | 020117 | |
| 3207 | 014632 | 042042 | 051511 | 041101 | |
| 3208 | 014640 | 042514 | 006442 | 012 | |
| 3209 | 014645 | 105 | 052116 | 051105 | .ASCIZ /ENTER THE WATCH DOG ADDRESS/(15)<12> |
| 3210 | 014652 | 052040 | 042510 | 053440 | |
| 3211 | 014660 | 052101 | 044103 | 042040 | |
| 3212 | 014666 | 043517 | 040440 | 042104 | |
| 3213 | 014674 | 042522 | 051523 | 005015 | |
| 3214 | 014702 | 000 | | | |
| 3215 | | | | | |
| 3216 | 014703 | 124 | 040522 | 020120 | DNOCLK: .ASCII /TRAP TO 4 WHEN ATTEMPTING TO ADDRESS REAL TIME/(15)<12> |
| 3217 | 014710 | 047524 | 032040 | 053440 | |
| 3218 | 014716 | 042510 | 020116 | 052101 | |
| 3219 | 014724 | 042524 | 050115 | 044524 | |
| 3220 | 014732 | 043516 | 052040 | 020117 | |
| 3221 | 014740 | 042101 | 051104 | 051505 | |
| 3222 | 014746 | 020123 | 042522 | 046101 | |
| 3223 | 014754 | 052040 | 046511 | 006505 | |
| 3224 | 014762 | 012 | | | |
| 3225 | 014763 | 103 | 047514 | 045503 | .ASCIZ /CLOCK CORRECT PROBLEM AND RE-START TEST/(15)<12> |
| 3226 | 014770 | 041440 | 051117 | 042522 | |
| 3227 | 014776 | 052103 | 050040 | 047522 | |
| 3228 | 015004 | 046102 | 046505 | 040440 | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| 3229 | 015012 | 042116 | 051040 | 026505 | |
| 3230 | 015020 | 052123 | 051101 | 020124 | |
| 3231 | 015026 | 042524 | 052123 | 005015 | |
| 3232 | 015034 | 000 | | | |
| 3233 | | | | | |
| 3234 | 015035 | 114 | 047111 | 020105 | DNORN: .ASCII /LINE CLOCK NOT RUNNING OR RUNNING SLOW/<15><12> |
| 3235 | 015042 | 046103 | 041517 | 020113 | |
| 3236 | 015050 | 047516 | 020124 | 052522 | |
| 3237 | 015056 | 047116 | 047111 | 020107 | |
| 3238 | 015064 | 051117 | 051040 | 047125 | |
| 3239 | 015072 | 044516 | 043516 | 051440 | |
| 3240 | 015100 | 047514 | 006527 | 012 | |
| 3241 | 015105 | 103 | 051117 | 042522 | .ASCIZ /CORRECT PROBLEM AND RE-START TEST/<15><12> |
| 3242 | 015112 | 052103 | 050040 | 047522 | |
| 3243 | 015120 | 046102 | 046505 | 040440 | |
| 3244 | 015126 | 042116 | 051040 | 026505 | |
| 3245 | 015134 | 052123 | 051101 | 020124 | |
| 3246 | 015142 | 042524 | 052123 | 005015 | |
| 3247 | 015150 | 000 | | | |
| 3248 | | | | | |
| 3249 | 015151 | 124 | 040522 | 020120 | DNONW: .ASCIZ /TRAP TO 4 WHEN ATTEMPTING TO ADDRESS NW1!/<15><12> |
| 3250 | 015156 | 047524 | 032040 | 053440 | |
| 3251 | 015164 | 042510 | 020116 | 052101 | |
| 3252 | 015172 | 042524 | 050115 | 044524 | |
| 3253 | 015200 | 043516 | 052040 | 020117 | |
| 3254 | 015206 | 042101 | 051104 | 051505 | |
| 3255 | 015214 | 020123 | 053516 | 030461 | |
| 3256 | 015222 | 005015 | 000 | | |
| 3257 | | 015226 | | | |
| 3258 | 015226 | 005015 | 053516 | 030461 | DVECT: .EVEN .ASCIZ <15><12>/NW11 VECTOR IS:/ |
| 3259 | 015234 | 053040 | 041505 | 047524 | |
| 3260 | 015242 | 020122 | 051511 | 000072 | |
| 3261 | 015250 | 052524 | 047122 | 041440 | DSHTDN: .ASCIZ /TURN CPU POWER OFF-ON TO RESET NW11 TO STATIC STATE/<15><12> |
| 3262 | 015256 | 052520 | 050040 | 053517 | |
| 3263 | 015264 | 051105 | 047440 | 043106 | |
| 3264 | 015272 | 047455 | 020116 | 047524 | |
| 3265 | 015300 | 051040 | 051505 | 052105 | |
| 3266 | 015306 | 047040 | 030527 | 020061 | |
| 3267 | 015314 | 047524 | 051440 | 040524 | |
| 3268 | 015322 | 044524 | 020103 | 052123 | |
| 3269 | 015330 | 052101 | 006505 | 000012 | |
| 3270 | 015336 | 047502 | 052117 | 042440 | DNSEBT: .ASCIZ /BOOT ENABLE (BIT4,REG.0) DID NOT SET. RESET TESTS BY-PASSED/<15><12> |
| 3271 | 015344 | 040516 | 046102 | 020105 | |
| 3272 | 015352 | 041050 | 052111 | 026064 | |
| 3273 | 015360 | 042522 | 027107 | 024460 | |
| 3274 | 015366 | 042040 | 042111 | 047040 | |
| 3275 | 015374 | 052117 | 051440 | 052105 | |
| 3276 | 015402 | 020056 | 042522 | 042523 | |
| 3277 | 015410 | 020124 | 042524 | 052123 | |
| 3278 | 015416 | 020123 | 054502 | 050055 | |
| 3279 | 015424 | 051501 | 042523 | 006504 | |
| 3280 | 015432 | 000012 | | | |
| 3281 | 015434 | 052123 | 052101 | 051525 | EMI: .ASCIZ /STATUS REGISTER ERROR/<15><12> |
| 3282 | 015442 | 051040 | 043505 | 051511 | |
| 3283 | 015450 | 042524 | 020122 | 051105 | |
| 3284 | 015456 | 047522 | 006522 | 000012 | |

| | | | | | | |
|------|--------|--------|---------|--------|-------|---|
| 3285 | 015464 | 044524 | 042515 | 030440 | EM2: | .ASCIZ /TIME 1 OR TIME 2 FAILED TO SET/<15><12> |
| 3286 | 015472 | 047440 | 020122 | 044524 | | |
| 3287 | 015500 | 042515 | 031040 | 043040 | | |
| 3288 | 015506 | 044501 | 042514 | 020104 | | |
| 3289 | 015514 | 047524 | 051440 | 052105 | | |
| 3290 | 015522 | 005015 | 000 | | | |
| 3291 | 015525 | 105 | 051122 | 051117 | DH1: | .ASCIZ /ERROR PC, REG., GOOD, BAD/<15><12> |
| 3292 | 015532 | 050040 | 026103 | 051040 | | |
| 3293 | 015540 | 043505 | 026056 | 020040 | | |
| 3294 | 015546 | 047507 | 042117 | 020054 | | |
| 3295 | 015554 | 020040 | 040502 | 006504 | | |
| 3296 | 015562 | 000012 | | | | |
| 3297 | 015564 | 044524 | 042515 | 030440 | EM3: | .ASCIZ /TIME 1 OR TIME 2 FAILED TO RESET WITHIN 3 SEC/<15><12> |
| 3298 | 015572 | 047440 | 020122 | 044524 | | |
| 3299 | 015600 | 042515 | 031040 | 043040 | | |
| 3300 | 015606 | 044501 | 042514 | 020104 | | |
| 3301 | 015614 | 047524 | 051040 | 051505 | | |
| 3302 | 015622 | 052105 | 053440 | 052111 | | |
| 3303 | 015630 | 044510 | 020116 | 020063 | | |
| 3304 | 015636 | 042523 | 006503 | 000012 | | |
| 3305 | 015644 | 051105 | 047522 | 020122 | EM4: | .ASCIZ /ERROR !!TIME 1 EXCEEDS TIME 2/<15><12> |
| 3306 | 015652 | 020441 | 044524 | 042515 | | |
| 3307 | 015660 | 030440 | 042440 | 041530 | | |
| 3308 | 015666 | 042505 | 051504 | 052040 | | |
| 3309 | 015674 | 046511 | 020105 | 006462 | | |
| 3310 | 015702 | 000012 | | | | |
| 3311 | 015704 | 044524 | 042515 | 031040 | EM5: | .ASCIZ /TIME 2 TIME OUT DID NOT SET ALARM OR FRO FLOP/<15><12> |
| 3312 | 015712 | 052040 | 046511 | 020105 | | |
| 3313 | 015720 | 052517 | 020124 | 044504 | | |
| 3314 | 015726 | 020104 | 047516 | 020124 | | |
| 3315 | 015734 | 042523 | 020124 | 046101 | | |
| 3316 | 015742 | 051101 | 020115 | 051117 | | |
| 3317 | 015750 | 043040 | 047522 | 043040 | | |
| 3318 | 015756 | 047514 | 006520 | 000012 | | |
| 3319 | 015764 | 052523 | 050120 | 051117 | EM6: | .ASCIZ /SUPPORTING T1-T2 DID NOT CLEAR ALARM OR SET GREEN LITE/<15><12> |
| 3320 | 015772 | 044524 | 043516 | 052040 | | |
| 3321 | 016000 | 026461 | 031124 | 042040 | | |
| 3322 | 016006 | 042111 | 047040 | 052117 | | |
| 3323 | 016014 | 041440 | 042514 | 051101 | | |
| 3324 | 016022 | 040440 | 040514 | 046522 | | |
| 3325 | 016030 | 047440 | 020122 | 042523 | | |
| 3326 | 016036 | 020124 | 051107 | 042505 | | |
| 3327 | 016044 | 020116 | 044514 | 042524 | | |
| 3328 | 016052 | 005015 | 000 | | | |
| 3329 | 016055 | 127 | 052101 | 044103 | EM7: | .ASCIZ /WATCH DOG ENABLE WILL NOT SET/<15><12> |
| 3330 | 016062 | 042040 | 04...17 | 042440 | | |
| 3331 | 016070 | 040516 | 046102 | 020105 | | |
| 3332 | 016076 | 044527 | 046114 | 047040 | | |
| 3333 | 016104 | 052117 | 051440 | 052105 | | |
| 3334 | 016112 | 005015 | 000 | | | |
| 3335 | 016115 | 127 | 052101 | 044103 | EM10: | .ASCIZ /WATCH DOG ENABLE WILL NOT RESET/<15><12> |
| 3336 | 016122 | 042040 | 043517 | 042440 | | |
| 3337 | 016130 | 040516 | 046102 | 020105 | | |
| 3338 | 016136 | 044527 | 046114 | 047040 | | |
| 3339 | 016144 | 052117 | 051040 | 051505 | | |
| 3340 | 016152 | 052105 | 005015 | 000 | | |

M05

MAINDEC-11-DZNWA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 65
 DZNWA.SRC TRAP TABLE

| | | | | | | |
|------|--------|--------|--------|--------|-------|---|
| 3341 | 016157 | 122 | 041505 | 020126 | EM11: | .ASCIZ /RECV IN (BIT15,REG.0) FAILED TO SET OR RESET/<15><12> |
| 3342 | 016164 | 047111 | 024040 | 044502 | | |
| 3343 | 016172 | 030524 | 026065 | 042522 | | |
| 3344 | 016200 | 027107 | 024460 | 043040 | | |
| 3345 | 016206 | 044501 | 042514 | 020104 | | |
| 3346 | 016214 | 047524 | 051440 | 052105 | | |
| 3347 | 016222 | 047440 | 020122 | 042522 | | |
| 3348 | 016230 | 042523 | 006524 | 000012 | | |
| 3349 | 016236 | 040503 | 047116 | 052117 | EM12: | .ASCIZ /CANNOT CLEAR BOOT ENABLE/<15><12> |
| 3350 | 016244 | 041440 | 042514 | 051101 | | |
| 3351 | 016252 | 041040 | 047517 | 020124 | | |
| 3352 | 016260 | 047105 | 041101 | 042514 | | |
| 3353 | 016266 | 005015 | 000 | | | |
| 3354 | 016271 | 103 | 047101 | 047516 | EM13: | .ASCIZ /CANNOT SET BOOT ENABLE/<15><12> |
| 3355 | 016276 | 020124 | 042523 | 020124 | | |
| 3356 | 016304 | 047502 | 052117 | 042440 | | |
| 3357 | 016312 | 040516 | 046102 | 006505 | | |
| 3358 | 016320 | 000012 | | | | |
| 3359 | 016322 | 040503 | 047116 | 052117 | EM14: | .ASCIZ /CANNOT SET A GREEN OR AN AMBER LITE/<15><12> |
| 3360 | 016330 | 051440 | 052105 | 040440 | | |
| 3361 | 016336 | 043440 | 042522 | 047105 | | |
| 3362 | 016344 | 047440 | 020122 | 047101 | | |
| 3363 | 016352 | 040440 | 041115 | 051105 | | |
| 3364 | 016360 | 046040 | 052111 | 006505 | | |
| 3365 | 016366 | 000012 | | | | |
| 3366 | 016370 | 040503 | 047116 | 052117 | EM15: | .ASCIZ /CANNOT RESET A GREEN OR AN AMBER LITE/<15><12> |
| 3367 | 016376 | 051040 | 051505 | 052105 | | |
| 3368 | 016404 | 040440 | 043440 | 042522 | | |
| 3369 | 016412 | 047105 | 047440 | 020122 | | |
| 3370 | 016420 | 047101 | 040440 | 041115 | | |
| 3371 | 016426 | 051105 | 046040 | 052111 | | |
| 3372 | 016434 | 006505 | 000012 | | | |
| 3373 | 016440 | 047117 | 020105 | 042523 | EM16: | .ASCIZ /ONE SECOND CLOCK TIME 3 DID NOT SET OR RESET/<15><12> |
| 3374 | 016446 | 047503 | 042116 | 041440 | | |
| 3375 | 016454 | 047514 | 045503 | 052040 | | |
| 3376 | 016462 | 046511 | 020105 | 020063 | | |
| 3377 | 016470 | 044504 | 020104 | 047516 | | |
| 3378 | 016476 | 020124 | 042523 | 020124 | | |
| 3379 | 016504 | 051117 | 051040 | 051505 | | |
| 3380 | 016512 | 052105 | 005015 | 000 | | |
| 3381 | 016517 | 117 | 042516 | 051440 | EM17: | .ASCIZ /ONE SECOND CLOCK TIME 3 NOT WITHIN 10 PER CENT/<15><12> |
| 3382 | 016524 | 041505 | 047117 | 020104 | | |
| 3383 | 016532 | 046103 | 041517 | 020113 | | |
| 3384 | 016540 | 044524 | 042515 | 031440 | | |
| 3385 | 016546 | 047040 | 052117 | 053440 | | |
| 3386 | 016554 | 052111 | 044510 | 020116 | | |
| 3387 | 016562 | 030061 | 050040 | 051105 | | |
| 3388 | 016570 | 041440 | 047105 | 006524 | | |
| 3389 | 016576 | 000012 | | | | |
| 3390 | 016600 | 051105 | 047522 | 020122 | DH2: | .ASCIZ /ERROR PC, ITERATION, SECONDS, MILLISECONDS/<15><12> |
| 3391 | 016606 | 041520 | 020054 | 052111 | | |
| 3392 | 016614 | 051105 | 052101 | 047511 | | |
| 3393 | 016622 | 026116 | 051440 | 041505 | | |
| 3394 | 016630 | 047117 | 051504 | 020054 | | |
| 3395 | 016636 | 044515 | 046114 | 051511 | | |
| 3396 | 016644 | 041505 | 047117 | 051504 | | |

N05

MAINDEC-11-DZNWA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 66
 DZNWA.SRC TRAP TABLE

| | | | | | | |
|------|--------|--------|--------|--------|-------|---|
| 3397 | 016652 | 005015 | 000 | | | |
| 3398 | 016655 | 123 | 040524 | 044524 | EM20: | .ASCIZ /STATIC PANEL SWITCH CONDITIONS INCORRECT/<15><12> |
| 3399 | 016662 | 020103 | 040520 | 042516 | | |
| 3400 | 016670 | 020114 | 053523 | 052111 | | |
| 3401 | 016676 | 044103 | 041440 | 047117 | | |
| 3402 | 016704 | 044504 | 044524 | 047117 | | |
| 3403 | 016712 | 020123 | 047111 | 047503 | | |
| 3404 | 016720 | 051122 | 041505 | 006524 | | |
| 3405 | 016726 | 000012 | | | | |
| 3406 | 016730 | 042523 | 026524 | 042522 | EM21: | .ASCIZ /SET-RESET FAILURE IN DROUT BITS/<15><12> |
| 3407 | 016736 | 042523 | 020124 | 040506 | | |
| 3408 | 016744 | 046111 | 051125 | 020105 | | |
| 3409 | 016752 | 047111 | 042040 | 047522 | | |
| 3410 | 016760 | 052125 | 041040 | 052111 | | |
| 3411 | 016766 | 006523 | 000012 | | | |
| 3412 | 016772 | 042523 | 026524 | 042522 | EM22: | .ASCIZ /SET-RESET FAILURE WITH FRO FLOP (BIT0,REG.6)/<15><12> |
| 3413 | 017000 | 042523 | 020124 | 040506 | | |
| 3414 | 017006 | 046111 | 051125 | 020105 | | |
| 3415 | 017014 | 044527 | 044124 | 043040 | | |
| 3416 | 017022 | 047522 | 043040 | 047514 | | |
| 3417 | 017030 | 020120 | 041050 | 052111 | | |
| 3418 | 017036 | 026060 | 042522 | 027107 | | |
| 3419 | 017044 | 024466 | 005015 | 000 | | |
| 3420 | 017051 | 123 | 052105 | 051055 | FM23: | .ASCIZ /SET-RESET FAILURE WITH MINUTE FLOP(S)/<15><12> |
| 3421 | 017056 | 051505 | 052105 | 043040 | | |
| 3422 | 017064 | 044501 | 052514 | 042522 | | |
| 3423 | 017072 | 053440 | 052111 | 020110 | | |
| 3424 | 017100 | 044515 | 052516 | 042524 | | |
| 3425 | 017106 | 043040 | 047514 | 024120 | | |
| 3426 | 017114 | 024523 | 005015 | 000 | | |
| 3427 | 017121 | 061 | 020065 | 042523 | EM24: | .ASCIZ /15 SECOND FLOP (BIT1,REG.6) DID NOT SET ON A COUNT OF 15/<15><12> |
| 3428 | 017126 | 047503 | 042116 | 043040 | | |
| 3429 | 017134 | 047514 | 020120 | 041050 | | |
| 3430 | 017142 | 052111 | 026061 | 042522 | | |
| 3431 | 017150 | 027107 | 024466 | 042040 | | |
| 3432 | 017156 | 042111 | 047040 | 052117 | | |
| 3433 | 017164 | 051440 | 052105 | 047440 | | |
| 3434 | 017172 | 020116 | 020101 | 047503 | | |
| 3435 | 017200 | 047125 | 020124 | 043117 | | |
| 3436 | 017206 | 020440 | 006465 | 000012 | | |
| 3437 | 017214 | 032461 | 051440 | 041505 | EM25: | .ASCIZ /15 SECOND FLOP (BIT1,REG.6) DID NOT RESET ON A CCUNT OF 30/<15><12> |
| 3438 | 017222 | 047117 | 020104 | 046106 | | |
| 3439 | 017230 | 050117 | 024040 | 044502 | | |
| 3440 | 017236 | 030524 | 051054 | 043505 | | |
| 3441 | 017244 | 033056 | 020051 | 044504 | | |
| 3442 | 017252 | 020104 | 047516 | 020124 | | |
| 3443 | 017260 | 042522 | 042523 | 020124 | | |
| 3444 | 017266 | 047117 | 040440 | 041440 | | |
| 3445 | 017274 | 052517 | 052116 | 047440 | | |
| 3446 | 017302 | 020106 | 030063 | 005015 | | |
| 3447 | 017310 | 000 | | | | |
| 3448 | 017311 | 063 | 020060 | 042523 | EM26: | .ASCIZ /30 SECOND FLOP (BIT2,REG.6) DID NOT SET ON A COUNT OF 30/<15><12> |
| 3449 | 017316 | 047503 | 042116 | 043040 | | |
| 3450 | 017324 | 047514 | 020120 | 041050 | | |
| 3451 | 017332 | 052111 | 026062 | 042522 | | |
| 3452 | 017340 | 027107 | 024466 | 042040 | | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 3453 | 017346 | 042111 | 047040 | 052117 | |
| 3454 | 017354 | 051440 | 052105 | 047440 | |
| 3455 | 017362 | 020116 | 020101 | 047503 | |
| 3456 | 017370 | 047125 | 020124 | 043117 | |
| 3457 | 017376 | 031440 | 006460 | 000012 | |
| 3458 | 017404 | 030063 | 051440 | 041505 | EM27: .ASCIZ /30 SECOND FLOP (BIT2,REG.6) DID NOT RESET ON A COUNT OF 61/<15><12> |
| 3459 | 017412 | 047117 | 020104 | 046106 | |
| 3460 | 017420 | 050117 | 024040 | 044502 | |
| 3461 | 017426 | 031124 | 051054 | 043505 | |
| 3462 | 017434 | 033056 | 020051 | 044504 | |
| 3463 | 017442 | 020104 | 047516 | 020124 | |
| 3464 | 017450 | 042522 | 042523 | 020124 | |
| 3465 | 017456 | 047117 | 040440 | 041440 | |
| 3466 | 017464 | 052517 | 052116 | 047440 | |
| 3467 | 017472 | 020106 | 030466 | 005015 | |
| 3468 | 017500 | 000 | | | |
| 3469 | 017501 | 061 | 020065 | 051117 | EM30: .ASCIZ /15 OR 30 SECOND FLOP(S) WAS NOT RESET BY CLR 1/<15><12> |
| 3470 | 017506 | 031440 | 020060 | 042523 | |
| 3471 | 017514 | 047503 | 042116 | 043040 | |
| 3472 | 017522 | 047514 | 024120 | 024523 | |
| 3473 | 017530 | 053440 | 051501 | 047040 | |
| 3474 | 017536 | 052117 | 051040 | 051505 | |
| 3475 | 017544 | 052105 | 041040 | 020131 | |
| 3476 | 017552 | 046103 | 020122 | 006461 | |
| 3477 | 017560 | 000012 | | | |
| 3478 | 017562 | 044524 | 042515 | 031440 | EM31: .ASCIZ /TIME 3 (1 SEC CLOCK) WILL NOT TOGGLE 15 SEC. COUNTER/<15><12> |
| 3479 | 017570 | 024040 | 020061 | 042523 | |
| 3480 | 017576 | 020103 | 046103 | 041517 | |
| 3481 | 017604 | 024513 | 053440 | 046111 | |
| 3482 | 017612 | 020114 | 047516 | 020124 | |
| 3483 | 017620 | 047524 | 043507 | 042514 | |
| 3484 | 017626 | 030440 | 020065 | 042523 | |
| 3485 | 017634 | 027103 | 041440 | 052517 | |
| 3486 | 017642 | 052116 | 051105 | 005015 | |
| 3487 | 017650 | 000 | | | |
| 3488 | 017651 | 103 | 047101 | 047040 | EM32: .ASCIZ /CAN NOT SET BOOT (BIT3,REG.6)/<15><12> |
| 3489 | 017656 | 052117 | 051440 | 052105 | |
| 3490 | 017664 | 041040 | 047517 | 020124 | |
| 3491 | 017672 | 041050 | 052111 | 026063 | |
| 3492 | 017700 | 042522 | 027107 | 024466 | |
| 3493 | 017706 | 005015 | 000 | | |
| 3494 | 017711 | 103 | 047101 | 047040 | EM33: .ASCIZ /CAN NOT SET ALL MINUTE FLOPS BY SERIAL INPUT/<15><12> |
| 3495 | 017716 | 052117 | 051440 | 052105 | |
| 3496 | 017724 | 040440 | 046114 | 046440 | |
| 3497 | 017732 | 047111 | 052125 | 020105 | |
| 3498 | 017740 | 046106 | 050117 | 020123 | |
| 3499 | 017746 | 054502 | 051440 | 051105 | |
| 3500 | 017754 | 040511 | 020114 | 047111 | |
| 3501 | 017762 | 052520 | 006524 | 000012 | |
| 3502 | 017770 | 054105 | 051124 | 020101 | EM34: .ASCIZ /EXTRA T-1 PULSES DID NOT CAUSE ALARM/<15><12> |
| 3503 | 017776 | 026524 | 020061 | 052520 | |
| 3504 | 020004 | 051514 | 051505 | 042040 | |
| 3505 | 020012 | 042111 | 047040 | 052117 | |
| 3506 | 020020 | 041440 | 052501 | 042523 | |
| 3507 | 020026 | 040440 | 040514 | 046522 | |
| 3508 | 020034 | 005015 | 000 | | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 3509 | | | | | |
| 3510 | 020037 | 104 | 047522 | 052125 | DDR1: .ASCIZ /DROUT 1 DID NOT RESET, EXIT TEST 10/(15)(12) |
| 3511 | 020044 | 030440 | 042040 | 042111 | |
| 3512 | 020052 | 047040 | 052117 | 051040 | |
| 3513 | 020060 | 051505 | 052105 | 020054 | |
| 3514 | 020066 | 054105 | 052111 | 052040 | |
| 3515 | 020074 | 051505 | 020124 | 030061 | |
| 3516 | 020102 | 005015 | 000 | | |
| 3517 | 020105 | 114 | 043111 | 020124 | DLMPTS: .ASCIZ /LIFT "LAMP TEST" SWITCH-VERIFY ALL LAMPS LIT/(15)(12) |
| 3518 | 020112 | 046042 | 046501 | 020120 | |
| 3519 | 020120 | 042524 | 052123 | 020042 | |
| 3520 | 020126 | 053523 | 052111 | 044103 | |
| 3521 | 020134 | 053055 | 051105 | 043111 | |
| 3522 | 020142 | 020131 | 046101 | 020114 | |
| 3523 | 020150 | 040514 | 050115 | 020123 | |
| 3524 | 020156 | 044514 | 006524 | 000012 | |
| 3525 | 020164 | 040520 | 042516 | 020114 | DLPSW: .ASCII /PANEL LAMPS = PANEL SWITCHES/(15)(12) |
| 3526 | 020172 | 040514 | 050115 | 020123 | |
| 3527 | 020200 | 020075 | 040520 | 042516 | |
| 3528 | 020206 | 020114 | 053523 | 052111 | |
| 3529 | 020214 | 044103 | 051505 | 005015 | |
| 3530 | 020222 | 026503 | 020122 | 052040 | DCRCNT: .ASCIZ /C-R TO EXIT TEST/(15)(12) |
| 3531 | 020230 | 020117 | 054105 | 052111 | |
| 3532 | 020236 | 052040 | 051505 | 006524 | |
| 3533 | 020244 | 000012 | | | |
| 3534 | | | | | |
| 3535 | 020246 | 052524 | 047122 | 021040 | DPNLK1: .ASCIZ /TURN "ENTER SWITCHLOCK" TO "ENABLE" POSITION/(15)(12) |
| 3536 | 020254 | 047105 | 042524 | 020122 | |
| 3537 | 020262 | 053523 | 052111 | 044103 | |
| 3538 | 020270 | 047514 | 045503 | 020042 | |
| 3539 | 020276 | 047524 | 047140 | 047105 | |
| 3540 | 020304 | 041101 | 042514 | 020042 | |
| 3541 | 020312 | 047520 | 044521 | 044524 | |
| 3542 | 020320 | 047117 | 005015 | 000 | |
| 3543 | 020325 | 124 | 051125 | 020116 | DPNLK: .ASCIZ /TURN "ENTER SWITCHLOCK" TO "DISABLE" POSITION/(15)(12) |
| 3544 | 020332 | 042442 | 052116 | 051105 | |
| 3545 | 020340 | 051440 | 044527 | 041524 | |
| 3546 | 020346 | 046110 | 041517 | 021113 | |
| 3547 | 020354 | 052040 | 020117 | 042042 | |
| 3548 | 020362 | 051511 | 041101 | 042514 | |
| 3549 | 020370 | 020042 | 047520 | 044523 | |
| 3550 | 020376 | 044524 | 047117 | 005015 | |
| 3551 | 020404 | 000 | | | |
| 3552 | 020405 | 123 | 044527 | 041524 | DPLKER: .ASCIZ /SWITCHES NOT LOCKED!/(15)(12) |
| 3553 | 020412 | 042510 | 020123 | 047516 | |
| 3554 | 020420 | 020124 | 047514 | 045503 | |
| 3555 | 020426 | 042105 | 006441 | 000012 | |
| 3556 | 020434 | 044514 | 052106 | 021040 | DPNINT: .ASCIZ /LIFT "ENTER SWITCHES" SWITCH MOMENTARILY AND WAIT!/(15)(12) |
| 3557 | 020442 | 047105 | 042524 | 020122 | |
| 3558 | 020450 | 053523 | 052111 | 044103 | |
| 3559 | 020456 | 051505 | 020042 | 053523 | |
| 3560 | 020464 | 052111 | 044103 | 046440 | |
| 3561 | 020472 | 046517 | 047105 | 040524 | |
| 3562 | 020500 | 044522 | 054514 | 040440 | |
| 3563 | 020506 | 042116 | 053440 | 044501 | |
| 3564 | 020514 | 020524 | 005015 | 000 | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 3565 | 020521 | 116 | 020117 | 047111 | DNOINT: .ASCIZ /NO INTERRUPT FROM "ENTER SWITCHES" SWITCH/<15><12> |
| 3566 | 020526 | 042524 | 051122 | 050125 | |
| 3567 | 020534 | 020124 | 051106 | 046517 | |
| 3568 | 020542 | 021040 | 047105 | 042524 | |
| 3569 | 020550 | 020122 | 053523 | 052111 | |
| 3570 | 020556 | 044103 | 051505 | 020042 | |
| 3571 | 020564 | 053523 | 052111 | 044103 | |
| 3572 | 020572 | 005015 | 000 | | |
| 3573 | 020575 | 115 | 046125 | 044524 | DMULNT: .ASCIZ /MULTIPLE INTERRUPTS FROM "ENTER SWITCHES" SWITCH,COUNT =/ |
| 3574 | 020602 | 046120 | 020105 | 047111 | |
| 3575 | 020610 | 042524 | 051122 | 050125 | |
| 3576 | 020616 | 051524 | 043040 | 047522 | |
| 3577 | 020624 | 020115 | 042442 | 052116 | |
| 3578 | 020632 | 051105 | 051440 | 044527 | |
| 3579 | 020640 | 041524 | 042510 | 021123 | |
| 3580 | 020646 | 051440 | 044527 | 041524 | |
| 3581 | 020654 | 026110 | 047503 | 047125 | |
| 3582 | 020662 | 020124 | 000075 | | |
| 3583 | 020666 | 040442 | 040514 | 046522 | DWDEON: .ASCIZ /"ALARM-LOAD" SWITCH TO "ALARM ENABLE" POSITION/<15><12> |
| 3584 | 020674 | 046055 | 040517 | 021104 | |
| 3585 | 020702 | 051440 | 044527 | 041524 | |
| 3586 | 020710 | 020110 | 047524 | 021040 | |
| 3587 | 020716 | 046101 | 051101 | 020115 | |
| 3588 | 020724 | 047105 | 041101 | 042514 | |
| 3589 | 020732 | 020042 | 047520 | 044523 | |
| 3590 | 020740 | 044524 | 047117 | 005015 | |
| 3591 | 020746 | 000 | | | |
| 3592 | 020747 | 124 | 051125 | 020116 | DWDEOF: .ASCIZ /TURN "ALARM-LOAD" SWITCH TO "DISABLE"/<15><12> |
| 3593 | 020754 | 040442 | 040514 | 046522 | |
| 3594 | 020762 | 046055 | 040517 | 021104 | |
| 3595 | 020770 | 051440 | 044527 | 041524 | |
| 3596 | 020776 | 020110 | 047524 | 021040 | |
| 3597 | 021004 | 044504 | 040523 | 046102 | |
| 3598 | 021012 | 021105 | 005015 | 000 | |
| 3599 | 021017 | 111 | 044516 | 044524 | EM35: .ASCIZ /INITIALIZE DID NOT CLEAR FRO FLOP/<15><12> |
| 3600 | 021024 | 046101 | 055111 | 020105 | |
| 3601 | 021032 | 044504 | 020104 | 047516 | |
| 3602 | 021040 | 020124 | 046103 | 040505 | |
| 3603 | 021046 | 020122 | 051106 | 020117 | |
| 3604 | 021054 | 046106 | 050117 | 005015 | |
| 3605 | 021062 | 000 | | | |
| 3606 | 021063 | 102 | 047517 | 020124 | EM36: .ASCIZ /BOOT ONE-SHOT DID NOT CLEAR WITHIN 2 SEC./<15><12> |
| 3607 | 021070 | 047117 | 026505 | 044123 | |
| 3608 | 021076 | 052117 | 042040 | 042111 | |
| 3609 | 021104 | 047040 | 052117 | 041440 | |
| 3610 | 021112 | 042514 | 051101 | 053440 | |
| 3611 | 021120 | 052111 | 044510 | 020116 | |
| 3612 | 021125 | 020062 | 042523 | 027103 | |
| 3613 | 021134 | 005015 | 000 | | |
| 3614 | 021137 | 124 | 046511 | 020105 | EM37: .ASCIZ /TIME 1 DID NOT INTERRUPT WITHIN 48 M.S./<15><12> |
| 3615 | 021144 | 020061 | 044504 | 020104 | |
| 3616 | 021152 | 047516 | 020124 | 047111 | |
| 3617 | 021160 | 042524 | 051122 | 050125 | |
| 3618 | 021166 | 020124 | 044527 | 044124 | |
| 3619 | 021174 | 047111 | 032040 | 020070 | |
| 3620 | 021202 | 027115 | 027123 | 005015 | |

E06

MAINDEC-11-DZNA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 70
 DZNA.SRC TRAP TABLE

| | | | | | |
|------|--------|--------|--------|--------|---|
| 3621 | 021210 | 000 | | | |
| 3622 | 021211 | 102 | 047517 | 020124 | EM40: .ASCIZ /BOOT ENABLE DID NOT SET TIME 3 ENABLE/<15><12> |
| 3623 | 021216 | 047105 | 041101 | 042514 | |
| 3624 | 021224 | 042040 | 042111 | 047040 | |
| 3625 | 021232 | 052117 | 051440 | 052105 | |
| 3626 | 021240 | 052040 | 046511 | 020105 | |
| 3627 | 021246 | 020063 | 047105 | 041101 | |
| 3628 | 021254 | 042514 | 005015 | 000 | |
| 3629 | 021261 | 015 | 051012 | 026505 | DBTCND: .ASCII <15><12>/RE-BOOT GENERATED, TIME WAS: /<15><12> |
| 3630 | 021266 | 047502 | 052117 | 043440 | |
| 3631 | 021274 | 047105 | 051105 | 052101 | |
| 3632 | 021302 | 042105 | 052054 | 046511 | |
| 3633 | 021310 | 020105 | 040527 | 035123 | |
| 3634 | 021316 | 005015 | | | |
| 3635 | 021320 | 020040 | 044515 | 027116 | .ASCIZ / MIN. SEC. /<15><12> |
| 3636 | 021326 | 020040 | 042523 | 027103 | |
| 3637 | 021334 | 005015 | 000 | | |
| 3638 | 021337 | 101 | 020116 | 046101 | DALNBT: .ASCIZ /AN ALARM DID NOT CAUSE A BOOT TO OCCUR-ALARM BOOT JUMPER OUT/<15><12> |
| 3639 | 021344 | 051101 | 020115 | 044504 | |
| 3640 | 021352 | 020104 | 047516 | 020124 | |
| 3641 | 021360 | 040503 | 051525 | 020105 | |
| 3642 | 021366 | 020101 | 047502 | 052117 | |
| 3643 | 021374 | 052040 | 020117 | 041517 | |
| 3644 | 021402 | 052503 | 026522 | 046101 | |
| 3645 | 021410 | 051101 | 020115 | 047502 | |
| 3646 | 021416 | 052117 | 045040 | 046525 | |
| 3647 | 021424 | 042520 | 020122 | 052517 | |
| 3648 | 021432 | 006524 | 000012 | | |
| 3649 | 021436 | 046101 | 051101 | 020115 | DALBT: .ASCIZ /ALARM CAUSED A BOOT TO OCCUR-ALARM BOOT JUMPER IN/<15><12> |
| 3650 | 021444 | 040503 | 051525 | 042105 | |
| 3651 | 021452 | 040440 | 041040 | 047517 | |
| 3652 | 021460 | 020124 | 047524 | 047440 | |
| 3653 | 021466 | 041503 | 051125 | 040455 | |
| 3654 | 021474 | 040514 | 046527 | 041040 | |
| 3655 | 021502 | 047517 | 020124 | 052512 | |
| 3656 | 021510 | 050115 | 051105 | 044440 | |
| 3657 | 021516 | 006516 | 000012 | | |
| 3658 | 021522 | 040527 | 041524 | 020110 | DNOWDE: .ASCIZ /WATCH DOG ENABLE (BIT1,REG.0) NOT SET/<15><12> |
| 3659 | 021530 | 047504 | 020107 | 047105 | |
| 3660 | 021536 | 041101 | 042514 | 024040 | |
| 3661 | 021544 | 044502 | 030524 | 051054 | |
| 3662 | 021552 | 043505 | 030056 | 020051 | |
| 3663 | 021560 | 047516 | 020124 | 042523 | |
| 3664 | 021566 | 006524 | 000012 | | |
| 3665 | 021572 | 040527 | 041524 | 020110 | DALWDE: .ASCIZ /WATCH DOG ENABLE (BIT1,REG.0) NOT CLEARED/<15><12> |
| 3666 | 021600 | 047504 | 020107 | 047105 | |
| 3667 | 021606 | 041101 | 042514 | 024040 | |
| 3668 | 021614 | 044502 | 030524 | 051054 | |
| 3669 | 021622 | 043505 | 030056 | 020051 | |
| 3670 | 021630 | 047516 | 020124 | 046103 | |
| 3671 | 021636 | 040505 | 042522 | 006504 | |
| 3672 | 021644 | 000012 | | | |
| 3673 | 021646 | 052524 | 047122 | 021040 | DBTENA: .ASCII /TURN "ALARM-LOAD" SWITCH TO "LOAD ENABLE" AND/<15><12> |
| 3674 | 021654 | 046101 | 051101 | 026515 | |
| 3675 | 021662 | 047514 | 042101 | 020042 | |
| 3676 | 021670 | 053523 | 052111 | 044103 | |

| | | | | |
|------|--------|--------|--------|--------|
| 3677 | 021676 | 052040 | 020117 | 046042 |
| 3678 | 021704 | 040517 | 020104 | 047105 |
| 3679 | 021712 | 041101 | 042514 | 020042 |
| 3680 | 021720 | 047101 | 006504 | 012 |
| 3681 | 021725 | 040 | 047515 | 042515 |
| 3682 | 021732 | 052116 | 051101 | 046111 |
| 3683 | 021740 | 024131 | 020441 | 020051 |
| 3684 | 021746 | 044514 | 052106 | 021040 |
| 3685 | 021754 | 047514 | 042101 | 051442 |
| 3686 | 021762 | 044527 | 041524 | 006510 |
| 3687 | 021770 | 000012 | | |
| 3688 | 021772 | 046042 | 040517 | 021104 |
| 3689 | 022000 | 051440 | 044527 | 041524 |
| 3690 | 022006 | 020110 | 044504 | 020104 |
| 3691 | 022014 | 047516 | 020124 | 040503 |
| 3692 | 022022 | 051525 | 020105 | 020101 |
| 3693 | 022030 | 047502 | 052117 | 042440 |
| 3694 | 022036 | 040516 | 046102 | 024105 |
| 3695 | 022044 | 044502 | 032124 | 051054 |
| 3696 | 022052 | 043505 | 030056 | 006451 |
| 3697 | 022060 | 000012 | | |
| 3698 | 022062 | 044506 | 046105 | 020104 |
| 3699 | 022070 | 042522 | 040514 | 020131 |
| 3700 | 022076 | 042502 | 047111 | 020107 |
| 3701 | 022104 | 047524 | 043507 | 042514 |
| 3702 | 022112 | 020104 | 020101 | 020065 |
| 3703 | 022120 | 042523 | 027103 | 051040 |
| 3704 | 022126 | 052101 | 026105 | 026503 |
| 3705 | 022134 | 020122 | 047524 | 042440 |
| 3706 | 022142 | 044530 | 006524 | 000012 |
| 3707 | 022150 | 051104 | 052517 | 020124 |
| 3708 | 022156 | 046106 | 050117 | 020123 |
| 3709 | 022164 | 042502 | 047111 | 020107 |
| 3710 | 022172 | 051042 | 050111 | 046120 |
| 3711 | 022200 | 042105 | 020042 | 052101 |
| 3712 | 022206 | 030440 | 051440 | 041505 |
| 3713 | 022214 | 020056 | 047111 | 042524 |
| 3714 | 022222 | 053122 | 046101 | 026123 |
| 3715 | 022230 | 026503 | 020122 | 047524 |
| 3716 | 022236 | 042440 | 044530 | 006524 |
| 3717 | 022244 | 000012 | | |
| 3718 | 022246 | 041523 | 050117 | 047111 |
| 3719 | 022254 | 020107 | 047514 | 050117 |
| 3720 | 022262 | 020123 | 047506 | 020122 |
| 3721 | 022270 | 046524 | 026061 | 044124 |
| 3722 | 022276 | 047105 | 052040 | 031115 |
| 3723 | 022304 | 020754 | 026503 | 020122 |
| 3724 | 022312 | 047524 | 042440 | 044530 |
| 3725 | 022320 | 020124 | 047111 | 044504 |
| 3726 | 022326 | 044526 | 052504 | 046101 |
| 3727 | 022334 | 046040 | 047517 | 051520 |
| 3728 | 022342 | 005015 | 000 | |
| 3729 | 022345 | 105 | 042116 | 047440 |
| 3730 | 022352 | 020106 | 054503 | 046103 |
| 3731 | 022360 | 041511 | 052040 | 051505 |
| 3732 | 022366 | 051524 | 005015 | 000 |

.ASCIZ / MOMENTARILY(!!) LIFT "LOAD" SWITCH/<15><12>

DNOPNB: .ASCIZ /"LOAD" SWITCH DID NOT CAUSE A BOOT ENABLE(BIT4,REG.0)/<15><12>

DCYCFR: .ASCIZ /FIELD RELAY BEING TOGGLED A 5 SEC. RATE,C-R TO EXIT/<15><12>

DCYCDR: .ASCIZ /DROUT FLOPS BEING "RIPPLED" AT 1 SEC. INTERVALS,C-R TO EXIT/<15><12>

DT1T2: .ASCIZ /SCOPING LOOPS FOR TM1, THEN TM2, C-R TO EXIT INDIVIDUAL LOOPS/<15><12>

DEND: .ASCIZ /END OF CYCLIC TESTS/<15><12>

G06

MAINDEC-11-DZNWA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 72
DZNWA.SRC TRAP TABLE

3733
3734

000001

.END

N06

MAINDEC-11-DZNA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 80
 DZNA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

| | | | | | | | | | | | | | |
|---------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 2819 | 2825 | 2827* | 2829* | 2837* | 2841 | 2845 | 2846 | 2850 | 2865* | 2870* | 2889* | 2895* |
| | 2899* | 2906 | 2907 | 2937* | 2938 | 2939 | 2940* | 2945* | 2946* | 2947* | 2953 | 2978 | 2979 |
| | 2980 | 2981* | 2982* | 2999* | 3000* | 3001* | 3002* | 3003* | 3005 | 3008* | 3016* | 3017 | 3019 |
| | 3020* | 3022 | 3023 | 3024 | 3041* | 3042* | 3043* | 3044* | 3045* | 3046* | 3047 | 3050* | 3063 |
| | 3065* | 3067 | 3077 | 3079 | 3081 | 3082 | 3083 | 3084 | 3085 | 3087* | 3088* | 3101* | 3102* |
| | 3103* | 3104* | 3105* | 3106* | 3107* | 3108 | 3116* | 3120 | 3121 | 3122 | 3123 | 3124 | 3125 |
| | 3126 | 3146* | 3147 | | | | | | | | | | |
| SSTART | 001752 | | | | | | | | | | | | |
| STACK = | 001100 | 1162 | 1164# | | | | | | | | | | |
| START | 000200 | 648# | 1171 | | | | | | | | | | |
| STKLMT= | 177774 | 775# | | | | | | | | | | | |
| SUPXT | 007624 | 659# | | | | | | | | | | | |
| SUP12 | 007440 | 2147 | 2151# | | | | | | | | | | |
| SNEXT | 005100 | 2122 | 2124# | | | | | | | | | | |
| SNLMP | 010002 | 1696 | 1701# | | | | | | | | | | |
| SWR | 001140 | 2199 | 2201# | | | | | | | | | | |
| | | 805# | 1169 | 1191* | 1193 | 1199* | 2151 | 2596 | 2610 | 2612 | 2618 | 2625 | 2664 |
| | | 2672 | 3107 | 3120* | | | | | | | | | 2669 |
| SWREC | 000176 | 763# | 1199 | | | | | | | | | | |
| SWSTAT | 005042 | 1691 | 1693# | | | | | | | | | | |
| SW0 | = 000001 | 712# | | | | | | | | | | | |
| SW00 | = 000001 | 702# | 712 | | | | | | | | | | |
| SW01 | = 000002 | 701# | 711 | | | | | | | | | | |
| SW02 | = 000004 | 700# | 710 | | | | | | | | | | |
| SW03 | = 000010 | 699# | 709 | | | | | | | | | | |
| SW04 | = 000020 | 698# | 708 | | | | | | | | | | |
| SW05 | = 000040 | 697# | 707 | | | | | | | | | | |
| SW06 | = 000100 | 696# | 706 | | | | | | | | | | |
| SW07 | = 000200 | 695# | 705 | | | | | | | | | | |
| SW08 | = 000400 | 694# | 704 | | | | | | | | | | |
| SW09 | = 001000 | 693# | 703 | | | | | | | | | | |
| SW1 | = 000002 | 711# | | | | | | | | | | | |
| SW10 | = 002000 | 692# | 2151 | | | | | | | | | | |
| SW11 | = 004000 | 691# | | | | | | | | | | | |
| SW12 | = 010000 | 690# | | | | | | | | | | | |
| SW13 | = 020000 | 689# | | | | | | | | | | | |
| SW14 | = 040000 | 688# | | | | | | | | | | | |
| SW15 | = 100000 | 687# | | | | | | | | | | | |
| SW2 | = 000004 | 710# | | | | | | | | | | | |
| SW3 | = 000010 | 709# | | | | | | | | | | | |
| SW4 | = 000020 | 708# | | | | | | | | | | | |
| SW5 | = 000040 | 707# | | | | | | | | | | | |
| SW6 | = 000100 | 706# | | | | | | | | | | | |
| SW7 | = 000200 | 705# | | | | | | | | | | | |
| SW8 | = 000400 | 704# | | | | | | | | | | | |
| SW9 | = 001000 | 703# | | | | | | | | | | | |
| SYNC3 | 011620 | 1792 | 1818 | 1849 | 1883 | 2032 | 2545# | | | | | | |
| SYNC3A | 011642 | 1931 | 1978 | 2049 | 2550# | | | | | | | | |
| TBITVE= | 000014 | 745# | | | | | | | | | | | |
| TCLOK | 011410 | 1247 | 2484# | | | | | | | | | | |
| TEMPA | 001630 | 1133# | 1238* | 1242* | 1664* | 1665 | 1667 | 1670 | 1672 | 1676 | 1678* | 1743* | 1753 |
| | | 1766 | 2228* | 2244 | 2250* | 2256* | 2258* | 2261 | 2135 | 2136 | 2423 | 2429 | 2527 |
| TGO | = 000001 | 1072# | 1382 | 1404 | 1430 | 1441 | 1602 | 2092 | | | | | |
| TIMCK | 002332 | 1232 | 1238# | | | | | | | | | | |
| TIME3 | 004542 | 1634 | 1636# | | | | | | | | | | |
| TKVEC | = 000060 | 752# | | | | | | | | | | | |
| TMI | = 000400 | 1080# | 1385 | 1401 | 1405 | 1442 | 1603 | 2424 | 2524 | 2528 | | | |

| | | | | | | | | | | | | | | | | | | |
|--------------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| SICNT | 001104 | 789# | 2061 | 2155 | 2629* | 2630 | 2632* | 2642 | | | | | | | | | | |
| SILLUP | 014274 | 3099 | 3115 | 3132# | | | | | | | | | | | | | | |
| SINTAG | 001135 | 803# | | | | | | | | | | | | | | | | |
| SITEMB | 001114 | 793# | 2663* | 2684 | 2867 | | | | | | | | | | | | | |
| SLF | 001166 | 819# | 2684 | 2770 | 2780 | 2856 | | | | | | | | | | | | |
| SLPADR | 001106 | 790# | 1185* | 1263* | 1299* | 1342* | 1378* | 1426* | 1465* | 1498* | 1527* | 1598* | 1634* | 1691* | | | | |
| | | 1711* | 1741* | 1779* | 1813* | 1845* | 1878* | 1922* | 1998* | 2027* | 2082* | 2122* | 2172* | 2199* | | | | |
| | | 2223* | 2276* | 2327* | 2367* | 2393* | 2419* | 2620* | 2635* | 2640 | 2642 | | | | | | | |
| SLPERR | 001110 | 791# | 1166* | 2620 | 2636* | 2642 | 2674 | | | | | | | | | | | |
| SMAL = ***** | U | 1203 | 2635 | 2669 | 2809 | | | | | | | | | | | | | |
| SMEN | 012753 | 2784# | | | | | | | | | | | | | | | | |
| SMWR | 012742 | 2782# | | | | | | | | | | | | | | | | |
| SMXCNT | 012210 | 2633 | 2642# | | | | | | | | | | | | | | | |
| SMULL | 001154 | 811# | 2827 | 2856 | | | | | | | | | | | | | | |
| SMWTST= | 000200 | 1260# | 1296# | 1339# | 1375# | 1423# | 1462# | 1495# | 1524# | 1595# | 1631# | 1688# | 1708# | 1738# | | | | |
| | | 1776# | 1810# | 1842# | 1875# | 1919# | 1995# | 2024# | 2079# | 2119# | 2169# | 2196# | 2220# | 2273# | | | | |
| | | 2324# | 2364# | 2390# | 2416# | | | | | | | | | | | | | |
| SOCNT | 013602 | 2944# | 2973* | 2986# | | | | | | | | | | | | | | |
| SOMODE | 013604 | 2939# | 2943* | 2948 | 2951* | 2962* | 2988# | | | | | | | | | | | |
| SOVER | 012174 | 2597 | 2613 | 2621 | 2631 | 2639# | | | | | | | | | | | | |
| SPASS | 001100 | 786# | 2153 | 2450* | 2451* | 2459 | 2472 | 2627 | 2643 | | | | | | | | | |
| SPOWER | 014302 | 3130 | 3135# | | | | | | | | | | | | | | | |
| SPWRON | 014134 | 1179 | 3099# | 3127 | | | | | | | | | | | | | | |
| SPWRMG | 014270 | 3130# | | | | | | | | | | | | | | | | |
| SPWRUP | 014206 | 3109 | 3115# | | | | | | | | | | | | | | | |
| SOLES | 001164 | 817# | 2684 | 2763 | 2780 | 2856 | | | | | | | | | | | | |
| SROCHR | 012344 | 2700# | 3169 | | | | | | | | | | | | | | | |
| SRODEC= | ***** | U | 3172 | | | | | | | | | | | | | | | |
| SROLIN | 012464 | 2728# | 3170 | | | | | | | | | | | | | | | |
| SROCT | 013606 | 2999# | 3171 | | | | | | | | | | | | | | | |
| SROSZ = | 000010 | 2721# | | | | | | | | | | | | | | | | |
| SRTNAD | 011364 | 2471# | | | | | | | | | | | | | | | | |
| SR2A = | ***** | U | 3172 | | | | | | | | | | | | | | | |
| SSAVRE= | ***** | U | 3172 | | | | | | | | | | | | | | | |
| SSAVR6 | 014300 | 3108# | 3116 | 3117* | 3118* | 3134# | | | | | | | | | | | | |
| SSCOPE | 011742 | 1173 | 2595# | | | | | | | | | | | | | | | |
| SSETUP= | 000037 | 1153# | 1172 | 1173 | 1175 | 1177 | 1179 | 1181 | 1182 | 1183 | 1185 | 2448 | 2596 | 2657 | | | | |
| | | 2672 | 2679 | 2689 | 2786 | | | | | | | | | | | | | |
| | | 1153# | | | | | | | | | | | | | | | | |
| SSTUP = | 177777 | 2605 | 2634# | | | | | | | | | | | | | | | |
| SSVLAD | 012146 | 768# | 773 | | | | | | | | | | | | | | | |
| SSVPC = | 000200 | 630 | 631# | 632# | 637 | 638 | 639 | 640 | 641 | 642 | 643 | 644 | 815 | 816 | | | | |
| SSWR = | 165400 | 817 | 1182 | 1183 | 1185 | 1186 | 1262 | 1298 | 1341 | 1377 | 1425 | 1464 | 1497 | 1526 | | | | |
| | | 1597 | 1633 | 1690 | 1710 | 1740 | 1778 | 1812 | 1844 | 1877 | 1921 | 1997 | 2026 | 2081 | 2121# | 2171* | 2198* | |
| | | 2121 | 2171 | 2198 | 2222 | 2275 | 2326 | 2366 | 2392 | 2418 | 2443 | 2449 | 2464 | 2470 | | | | |
| | | 2472 | 2587 | 2588 | 2589 | 2590 | 2591 | 2596 | 2608 | 2610 | 2611 | 2614 | 2615 | 2616 | | | | |
| | | 2623 | 2624 | 2625 | 2636 | 2639 | 2642 | 2649 | 2650 | 2651 | 2652 | 2660 | 2664 | 2669 | | | | |
| | | 2672 | 2684 | 3131 | | | | | | | | | | | | | | |
| SSWRMK= | 000000 | 644 | 645 | 2591 | 2592 | 2612 | | | | | | | | | | | | |
| STIMES | 001160 | 815# | 1182* | 1262* | 1298* | 1341* | 1377* | 1425* | 1464* | 1497* | 1526* | 1597* | 1633* | 1690* | | | | |
| | | 1710# | 1740* | 1778* | 1812* | 1844* | 1877* | 1921* | 1997* | 2026* | 2081* | 2121# | 2171* | 2198* | | | | |
| | | 2222* | 2275* | 2326* | 2366* | 2392* | 2418* | 2449* | 2623* | 2630 | 2633* | 2642 | | | | | | |
| STKB | 001146 | 808# | 2564 | 2687 | 2704 | 2710 | | | | | | | | | | | | |
| STKS | 001144 | 807# | 2562 | 2687 | 2702 | 2708 | | | | | | | | | | | | |
| STN = | 000037 | 630# | 1260 | 1262# | 1296 | 1298# | 1339 | 1341# | 1375 | 1377# | 1423 | 1425# | 1462 | 1464# | | | | |
| | | 1495 | 1497# | 1524 | 1526# | 1595 | 1597# | 1631 | 1633# | 1688 | 1690# | 1708 | 1710# | 1738 | | | | |

| | | | |
|--------|------|------|------|
| .SWRLO | 620# | 645# | |
| .SACT1 | 1# | 620# | 764 |
| .SAPT8 | 1# | | |
| .SAPTH | 1# | | |
| .SAPTY | 1# | | |
| .SASTA | 1# | | |
| .SCATC | 1# | 620# | 755 |
| .SCMTA | 1# | 620# | 778 |
| .SDB20 | 1# | | |
| .SDB20 | 1# | | |
| .SDIV | 1# | | |
| .SEOP | 1# | 620# | 2438 |
| .SERRO | 1# | 620# | 2643 |
| .SERRT | 1# | 620# | 2856 |
| .SFLT | 1# | | |
| .SPOWE | 1# | 620# | 3095 |
| .SRAND | 1# | | |
| .SRDDE | 1# | | |
| .SRDOC | 1# | 620# | 2989 |
| .SREAD | 1# | 620# | 2684 |
| .SR2AZ | 1# | | |
| .SSAVE | 1# | 620# | |
| .SSB20 | 1# | | |
| .SSB20 | 1# | | |
| .SSCOP | 1# | 620# | 2581 |
| .SSIZE | 1# | | |
| .SSUPR | 1# | | |
| .STRAP | 1# | 620# | 3138 |
| .STYP8 | 1# | | |
| .STYPD | 1# | 620# | 3028 |
| .STYPE | 1# | 620# | 2786 |
| .STYPO | 1# | 620# | 2912 |
| .S40CA | 1# | | |
| .1170 | 1# | | |

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| ADD | 1216 | 1218 | 1220 | 1283 | 1677 | 1965 | 1968 | 1975 | 2568 | 2813 | 2878 | 2940 | 2950 | 3017 | 3060 |
| ASL | 1799 | 2407 | 2875 | 2876 | 2877 | 3010 | 3012 | 3014 | 3150 | | | | | | |
| ASLB | 3065 | | | | | | | | | | | | | | |
| ASR | 1956 | 1957 | 1958 | | | | | | | | | | | | |
| BCC | 3066 | | | | | | | | | | | | | | |
| BEQ | 1204 | 1309 | 1314 | 1386 | 1447 | 1480 | 1505 | 1509 | 1533 | 1540 | 1562 | 1564 | 1581 | 1608 | 1620 |
| | 1647 | 1658 | 1663 | 1696 | 1718 | 1723 | 1760 | 1767 | 1787 | 1801 | 1830 | 1862 | 1887 | 1898 | 1906 |
| | 1939 | 1942 | 1964 | 1967 | 2042 | 2129 | 2152 | 2154 | 2245 | 2257 | 2259 | 2281 | 2302 | 2311 | 2350 |
| | 2381 | 2408 | 2463 | 2533 | 2546 | 2553 | 2563 | 2611 | 2613 | 2615 | 2619 | 2628 | 2658 | 2673 | 2676 |
| | 2749 | 2816 | 2851 | 2880 | 2885 | 2891 | 2902 | 2967 | 3009 | | | | | | |
| BGE | 2631 | | | | | | | | | | | | | | |
| BGT | 2454 | 2718 | 2974 | 3074 | | | | | | | | | | | |
| BHI | 1214 | 2057 | 2617 | | | | | | | | | | | | |
| BHIS | 1668 | 1896 | 1982 | | | | | | | | | | | | |
| BIC | 1312 | 1357 | 1477 | 1694 | 1755 | 1940 | 1974 | 1986 | 2009 | 2072 | 2095 | 2103 | 2243 | 2252 | 2382 |
| | 2451 | 2489 | 2514 | 2565 | 2576 | 2705 | 2711 | 2719 | 2964 | 3016 | | | | | |
| BIS | 1284 | 1329 | 1348 | 1353 | 1364 | 1381 | 1382 | 1404 | 1428 | 1430 | 1441 | 1467 | 1468 | 1478 | 1481 |
| | 1485 | 1499 | 1501 | 1529 | 1538 | 1553 | 1600 | 1602 | 1636 | 1714 | 1716 | 1744 | 1745 | 1782 | 1793 |
| | 1817 | 1848 | 1882 | 1924 | 2001 | 2017 | 2030 | 2034 | 2050 | 2090 | 2091 | 2092 | 2125 | 2135 | 2136 |
| | 2234 | 2330 | 2369 | 2372 | 2395 | 2421 | 2423 | 2429 | 2508 | 2509 | 2510 | 2527 | 2575 | 2969 | 2970 |
| | 3068 | 3069 | | | | | | | | | | | | | |
| BISB | 2867 | | | | | | | | | | | | | | |
| BIT | 1349 | 1385 | 1387 | 1401 | 1405 | 1407 | 1431 | 1433 | 1442 | 1446 | 1452 | 1469 | 1475 | 1479 | 1486 |
| | 1502 | 1508 | 1532 | 1539 | 1541 | 1554 | 1561 | 1563 | 1574 | 1580 | 1603 | 1607 | 1609 | 1615 | 1619 |
| | 1638 | 1646 | 1655 | 1657 | 1660 | 1662 | 1729 | 1747 | 1759 | 1764 | 1786 | 1794 | 1800 | 1821 | 1829 |
| | 1852 | 1861 | 1886 | 1889 | 1897 | 1905 | 1935 | 1949 | 1963 | 1966 | 2006 | 2010 | 2036 | 2041 | 2054 |
| | 2065 | 2102 | 2128 | 2137 | 2143 | 2146 | 2151 | 2178 | 2249 | 2280 | 2290 | 2301 | 2310 | 2335 | 2349 |
| | 2380 | 2424 | 2430 | 2436 | 2524 | 2528 | 2534 | 2545 | 2547 | 2550 | 2552 | 2562 | 2596 | 2610 | 2618 |
| | 2625 | 2664 | 2672 | | | | | | | | | | | | |
| BITB | 2639 | | | | | | | | | | | | | | |
| PLC | 1395 | 1415 | 1511 | 1557 | 1566 | 1641 | 1649 | 1673 | 1725 | 2013 | 2068 | 2094 | 2184 | 2242 | 2283 |
| | 2296 | 2304 | 2341 | 2376 | 2403 | 2744 | | | | | | | | | |
| BLOS | 2156 | 2732 | | | | | | | | | | | | | |
| BLT | 2716 | 2830 | 2975 | 3057 | 3073 | | | | | | | | | | |
| BMI | 3064 | | | | | | | | | | | | | | |
| BNE | 1170 | 1194 | 1241 | 1243 | 1311 | 1321 | 1350 | 1360 | 1363 | 1367 | 1388 | 1402 | 1406 | 1408 | 1432 |
| | 1434 | 1443 | 1445 | 1453 | 1470 | 1476 | 1487 | 1503 | 1542 | 1555 | 1575 | 1604 | 1606 | 1610 | 1616 |
| | 1639 | 1656 | 1661 | 1666 | 1671 | 1679 | 1730 | 1748 | 1754 | 1765 | 1795 | 1822 | 1853 | 1890 | 1936 |
| | 1950 | 2007 | 2011 | 2037 | 2044 | 2055 | 2062 | 2066 | 2138 | 2144 | 2147 | 2179 | 2291 | 2336 | 2347 |
| | 2425 | 2431 | 2437 | 2486 | 2513 | 2525 | 2529 | 2531 | 2535 | 2548 | 2551 | 2567 | 2597 | 2626 | 2665 |
| | 2680 | 2707 | 2713 | 2736 | 2738 | 2754 | 2758 | 2768 | 2810 | 2818 | 2826 | 2840 | 2847 | 2868 | 2894 |
| | 2965 | 3062 | 3119 | | | | | | | | | | | | |
| BPL | 2670 | 2703 | 2709 | 2804 | 2844 | 2963 | 3048 | 3078 | | | | | | | |
| BR | 1162 | 1196 | 1232 | 1267 | 1271 | 1322 | 1365 | 1400 | 1506 | 1537 | 1546 | 1568 | 1572 | 1577 | 1645 |
| | 1653 | 1669 | 1757 | 1791 | 1826 | 1857 | 1899 | 1903 | 1951 | 1979 | 1984 | 2070 | 2100 | 2181 | 2182 |
| | 2186 | 2205 | 2213 | 2239 | 2240 | 2246 | 2248 | 2251 | 2284 | 2293 | 2294 | 2298 | 2307 | 2308 | 2338 |
| | 2339 | 2343 | 2344 | 2378 | 2379 | 2383 | 2405 | 2406 | 2409 | 2427 | 2433 | 2599 | 2605 | 2608 | 2621 |
| | 2624 | 2714 | 2747 | 2756 | 2762 | 2764 | 2806 | 2823 | 2833 | 2842 | 2849 | 2873 | 2897 | 2904 | 2941 |
| | 2956 | 2977 | 3018 | 3059 | 3076 | 3111 | 3133 | | | | | | | | |
| CLR | 1163 | 1168 | 1182 | 1183 | 1239 | 1281 | 1306 | 1344 | 1345 | 1413 | 1552 | 1624 | 1659 | 1674 | 1675 |
| | 1699 | 1715 | 1727 | 1743 | 1762 | 1781 | 1789 | 1815 | 1816 | 1847 | 1881 | 1891 | 1892 | 1908 | 1927 |
| | 1962 | 1985 | 2008 | 2063 | 2084 | 2106 | 2107 | 2177 | 2201 | 2209 | 2228 | 2233 | 2235 | 2255 | 2278 |
| | 2289 | 2300 | 2329 | 2334 | 2345 | 2374 | 2384 | 2396 | 2400 | 2435 | 2448 | 2449 | 2497 | 2498 | 2499 |
| | 2500 | 2536 | 2577 | 2623 | 2637 | 2729 | 2752 | 2866 | 2954 | 3006 | 3007 | 3051 | 3054 | 3117 | |
| CLRB | 2487 | 2622 | 2759 | 2769 | 2822 | 2848 | 3080 | | | | | | | | |
| CMP | 1169 | 1193 | 1213 | 1308 | 1313 | 1320 | 1362 | 1366 | 1717 | 1941 | 2061 | 2155 | 2606 | 2630 | 2679 |

| | | | | | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| CMPB | 2706 | 2712 | 2715 | 2717 | 2731 | 2743 | 3072 | | | | | | | | |
| | 1394 | 1414 | 1504 | 1510 | 1556 | 1565 | 1640 | 1648 | 1667 | 1670 | 1672 | 1724 | 1895 | 1981 | 2012 |
| | 2043 | 2056 | 2067 | 2093 | 2183 | 2241 | 2282 | 2295 | 2303 | 2340 | 2346 | 2375 | 2402 | 2485 | 2566 |
| | 2612 | 2616 | 2735 | 2753 | 2757 | 2767 | 2815 | 2817 | 2825 | 2846 | 2850 | | | | |
| DEC | 1242 | 1444 | 1605 | 2256 | 2452 | 2512 | 2530 | 2742 | 2874 | | | | | | |
| DECB | 1678 | 2829 | 2832 | 2962 | 2973 | | | | | | | | | | |
| EMT | 649 | | | | | | | | | | | | | | |
| HALT | 761 | 1235 | 1245 | 1270 | 2476 | 2671 | 2681 | 2805 | 3110 | 3132 | | | | | |
| INC | 1756 | 2250 | 2258 | 2450 | 2523 | 2629 | 2660 | 2968 | 2976 | 3058 | 3118 | | | | |
| INCB | 2484 | 2488 | 2634 | 2657 | 2852 | | | | | | | | | | |
| IOT | 650 | | | | | | | | | | | | | | |
| JMP | 775 | 777 | 1205 | 1236 | 1246 | 2159 | 2470 | | | | | | | | |
| JSR | 1351 | 1380 | 1429 | 1474 | 1500 | 1530 | 1531 | 1551 | 1579 | 1637 | 1746 | 1784 | 1792 | 1802 | 1818 |
| | 1820 | 1828 | 1834 | 1849 | 1851 | 1860 | 1867 | 1880 | 1883 | 1885 | 1888 | 1894 | 1904 | 1925 | 1930 |
| | 1931 | 1934 | 1977 | 1978 | 2000 | 2002 | 2005 | 2031 | 2032 | 2033 | 2048 | 2049 | 2052 | 2064 | 2089 |
| | 2126 | 2133 | 2142 | 2176 | 2180 | 2204 | 2212 | 2229 | 2238 | 2285 | 2288 | 2292 | 2306 | 2331 | 2333 |
| MOV | 2337 | 2348 | 2371 | 2373 | 2377 | 2398 | 2404 | 2426 | 2432 | 2465 | 2666 | 2824 | 2831 | 2838 | |
| | 1161 | 1167 | 1171 | 1173 | 1174 | 1175 | 1176 | 1177 | 1178 | 1179 | 1180 | 1181 | 1185 | 1186 | 1189 |
| | 1190 | 1191 | 1192 | 1197 | 1199 | 1200 | 1201 | 1212 | 1215 | 1217 | 1219 | 1221 | 1228 | 1229 | 1230 |
| | 1238 | 1247 | 1248 | 1262 | 1263 | 1265 | 1273 | 1275 | 1280 | 1282 | 1298 | 1299 | 1304 | 1307 | 1315 |
| | 1316 | 1317 | 1341 | 1342 | 1347 | 1352 | 1354 | 1377 | 1378 | 1379 | 1389 | 1390 | 1391 | 1396 | 1397 |
| | 1398 | 1409 | 1410 | 1411 | 1425 | 1426 | 1435 | 1436 | 1437 | 1440 | 1448 | 1449 | 1450 | 1464 | 1465 |
| | 1471 | 1472 | 1482 | 1483 | 1497 | 1498 | 1512 | 1513 | 1526 | 1527 | 1534 | 1535 | 1543 | 1544 | 1547 |
| | 1548 | 1558 | 1559 | 1569 | 1570 | 1582 | 1583 | 1597 | 1598 | 1601 | 1611 | 1612 | 1621 | 1622 | 1633 |
| | 1634 | 1642 | 1643 | 1650 | 1651 | 1664 | 1690 | 1691 | 1693 | 1697 | 1698 | 1710 | 1711 | 1713 | 1719 |
| | 1720 | 1726 | 1740 | 1741 | 1749 | 1750 | 1761 | 1778 | 1779 | 1783 | 1785 | 1788 | 1796 | 1797 | 1812 |
| | 1813 | 1819 | 1823 | 1824 | 1827 | 1831 | 1832 | 1844 | 1845 | 1850 | 1854 | 1855 | 1859 | 1863 | 1864 |
| | 1877 | 1878 | 1884 | 1893 | 1900 | 1901 | 1907 | 1921 | 1922 | 1926 | 1929 | 1932 | 1933 | 1938 | 1943 |
| | 1944 | 1947 | 1953 | 1954 | 1959 | 1970 | 1976 | 1983 | 1997 | 1998 | 2003 | 2004 | 2014 | 2015 | 2026 |
| | 2027 | 2029 | 2035 | 2038 | 2039 | 2045 | 2046 | 2051 | 2053 | 2058 | 2059 | 2081 | 2082 | 2085 | 2086 |
| | 2087 | 2096 | 2097 | 2098 | 2104 | 2105 | 2108 | 2121 | 2122 | 2124 | 2130 | 2131 | 2139 | 2140 | 2148 |
| | 2149 | 2171 | 2172 | 2198 | 2199 | 2208 | 2211 | 2222 | 2223 | 2225 | 2226 | 2227 | 2236 | 2253 | 2254 |
| | 2261 | 2275 | 2276 | 2312 | 2313 | 2314 | 2326 | 2327 | 2351 | 2352 | 2353 | 2366 | 2367 | 2392 | 2393 |
| | 2399 | 2401 | 2418 | 2419 | 2455 | 2459 | 2462 | 2526 | 2601 | 2602 | 2604 | 2607 | 2620 | 2632 | 2633 |
| | 2635 | 2636 | 2639 | 2640 | 2659 | 2661 | 2674 | 2677 | 2700 | 2701 | 2728 | 2730 | 2741 | 2772 | 2773 |
| | 2774 | 2775 | 2807 | 2808 | 2812 | 2827 | 2865 | 2870 | 2879 | 2884 | 2889 | 2890 | 2892 | 2895 | 2899 |
| | 2906 | 2907 | 2937 | 2945 | 2946 | 2947 | 2953 | 2960 | 2978 | 2979 | 2980 | 2981 | 2982 | 2999 | 3000 |
| | 3001 | 3002 | 3003 | 3005 | 3020 | 3021 | 3022 | 3023 | 3024 | 3041 | 3042 | 3043 | 3044 | 3045 | 3046 |
| | 3047 | 3052 | 3055 | 3075 | 3081 | 3082 | 3083 | 3084 | 3085 | 3087 | 3088 | 3099 | 3100 | 3101 | 3102 |
| | 3103 | 3104 | 3105 | 3106 | 3107 | 3108 | 3109 | 3115 | 3116 | 3120 | 3121 | 3122 | 3123 | 3124 | 3125 |
| MOV8 | 3126 | 3127 | 3128 | 3146 | 3147 | 3151 | | | | | | | | | |
| | 1184 | 1676 | 2564 | 2638 | 2663 | 2704 | 2710 | 2734 | 2739 | 2745 | 2750 | 2765 | 2809 | 2837 | 2845 |
| | 2938 | 2939 | 2942 | 2943 | 2944 | 2948 | 2951 | 2952 | 2971 | 3008 | 3050 | 3053 | 3067 | 3070 | 3079 |
| | 3149 | | | | | | | | | | | | | | |
| NEG | 2949 | 3049 | | | | | | | | | | | | | |
| NOP | 1383 | 1384 | 1515 | 1586 | 1617 | 1618 | 1681 | 1701 | 1910 | 1937 | 2110 | 2127 | 2134 | 2145 | 2160 |
| | 2187 | 2264 | 2316 | 2410 | 2466 | 2467 | 2468 | | | | | | | | |
| RESET | 1758 | 2158 | 2309 | 2464 | | | | | | | | | | | |
| ROL | 2955 | 2957 | 2958 | 2959 | 2961 | 3011 | 3013 | 3015 | | | | | | | |
| RTI | 1198 | 2109 | 2237 | 2490 | 2641 | 2683 | 2720 | 2776 | 2814 | 2983 | 3025 | 3089 | 3131 | | |
| RTS | 2501 | 2515 | 2537 | 2549 | 2554 | 2569 | 2578 | 2854 | 2909 | 3152 | | | | | |
| SUB | 2662 | 3056 | | | | | | | | | | | | | |
| SWAB | 2210 | | | | | | | | | | | | | | |
| TRAP | 3154 | 3163 | 3164 | 3165 | 3166 | 3169 | 3170 | 3171 | | | | | | | |
| TST | 1203 | 1231 | 1240 | 1266 | 1310 | 1319 | 1358 | 1359 | 1361 | 1695 | 1722 | 1753 | 1766 | 1948 | 2153 |
| | 2244 | 2511 | 2532 | 2603 | 2627 | 2669 | 2675 | 2737 | 2748 | 2771 | 2811 | 2819 | 2841 | 2901 | 2966 |

| | | | | | | | | | | | | | | | |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| TSTB | 3019 | 3061 | 3071 | 3148 | | | | | | | | | | | |
| WAIT | 1665 | 2614 | 2702 | 2708 | 2803 | 2843 | 2893 | 3063 | 3077 | | | | | | |
| .ASCII | 1302 | | | | | | | | | | | | | | |
| .ASCIIZ | 817 | 818 | 3176 | 3184 | 3197 | 3216 | 3234 | 3525 | 3629 | 3673 | | | | | |
| | 819 | 2473 | 2780 | 2781 | 2782 | 2784 | 2910 | 3135 | 3209 | 3225 | 3241 | 3249 | 3258 | 3261 | 3270 |
| | 3281 | 3285 | 3291 | 3297 | 3305 | 3311 | 3319 | 3329 | 3335 | 3341 | 3349 | 3354 | 3359 | 3366 | 3373 |
| | 3381 | 3390 | 3398 | 3406 | 3412 | 3420 | 3427 | 3437 | 3448 | 3458 | 3469 | 3478 | 3488 | 3494 | 3502 |
| | 3510 | 3517 | 3530 | 3535 | 3543 | 3552 | 3556 | 3565 | 3573 | 3583 | 3592 | 3599 | 3606 | 3614 | 3622 |
| | 3635 | 3638 | 3649 | 3658 | 3665 | 3681 | 3688 | 3698 | 3707 | 3718 | 3729 | | | | |
| .BLKB | 2779 | | | | | | | | | | | | | | |
| .BLKW | 3094 | | | | | | | | | | | | | | |
| .BYTE | 787 | 788 | 793 | 794 | 802 | 803 | 811 | 812 | 813 | 814 | 1065 | 1067 | 1277 | 1278 | 2472 |
| | 2777 | 2778 | 2984 | 2985 | 2986 | 2987 | | | | | | | | | |
| .DSABL | 2689 | | | | | | | | | | | | | | |
| .ENABL | 1 | 620 | 2687 | | | | | | | | | | | | |
| .END | 3734 | | | | | | | | | | | | | | |
| .ENDC | 625 | 641 | 643 | 644 | 645 | 649 | 741 | 755 | 764 | 767 | 771 | 773 | 781 | 785 | 787 |
| | 615 | 816 | 817 | 821 | 1153 | 1156 | 1160 | 1171 | 1172 | 1175 | 1177 | 1179 | 1181 | 1182 | 1183 |
| | 1185 | 1187 | 1203 | 1224 | 1227 | 1251 | 1258 | 1261 | 1262 | 1263 | 1264 | 1278 | 1279 | 1282 | 1287 |
| | 1297 | 1298 | 1299 | 1300 | 1332 | 1338 | 1340 | 1341 | 1342 | 1343 | 1370 | 1375 | 1376 | 1377 | 1378 |
| | 1379 | 1418 | 1422 | 1424 | 1425 | 1426 | 1427 | 1456 | 1461 | 1463 | 1464 | 1465 | 1466 | 1490 | 1494 |
| | 1496 | 1497 | 1498 | 1499 | 1518 | 1523 | 1525 | 1526 | 1527 | 1528 | 1589 | 1594 | 1596 | 1597 | 1598 |
| | 1599 | 1627 | 1630 | 1632 | 1633 | 1634 | 1635 | 1684 | 1687 | 1689 | 1690 | 1691 | 1692 | 1704 | 1707 |
| | 1709 | 1710 | 1711 | 1712 | 1734 | 1737 | 1739 | 1740 | 1741 | 1742 | 1771 | 1775 | 1777 | 1778 | 1779 |
| | 1780 | 1806 | 1809 | 1811 | 1812 | 1813 | 1814 | 1838 | 1841 | 1843 | 1844 | 1845 | 1846 | 1871 | 1874 |
| | 1876 | 1877 | 1878 | 1879 | 1914 | 1918 | 1920 | 1921 | 1922 | 1923 | 1989 | 1994 | 1996 | 1997 | 1998 |
| | 1999 | 2019 | 2023 | 2025 | 2026 | 2027 | 2028 | 2075 | 2078 | 2080 | 2081 | 2082 | 2083 | 2113 | 2118 |
| | 2120 | 2121 | 2122 | 2123 | 2164 | 2168 | 2170 | 2171 | 2172 | 2173 | 2190 | 2195 | 2197 | 2198 | 2199 |
| | 2200 | 2216 | 2219 | 2221 | 2222 | 2223 | 2224 | 2268 | 2272 | 2274 | 2275 | 2276 | 2277 | 2319 | 2323 |
| | 2325 | 2326 | 2327 | 2328 | 2359 | 2363 | 2365 | 2366 | 2367 | 2368 | 2387 | 2389 | 2391 | 2392 | 2393 |
| | 2394 | 2413 | 2415 | 2417 | 2418 | 2419 | 2420 | 2441 | 2442 | 2443 | 2445 | 2448 | 2454 | 2457 | 2458 |
| | 2462 | 2464 | 2470 | 2472 | 2473 | 2476 | 2479 | 2483 | 2493 | 2496 | 2504 | 2518 | 2521 | 2541 | 2544 |
| | 2558 | 2561 | 2572 | 2574 | 2584 | 2587 | 2592 | 2596 | 2598 | 2609 | 2612 | 2613 | 2614 | 2616 | 2618 |
| | 2625 | 2629 | 2634 | 2635 | 2639 | 2642 | 2643 | 2646 | 2649 | 2657 | 2661 | 2666 | 2667 | 2668 | 2669 |
| | 2679 | 2683 | 2684 | 2687 | 2688 | 2689 | 2693 | 2721 | 2722 | 2730 | 2732 | 2735 | 2763 | 2780 | 2786 |
| | 2789 | 2809 | 2859 | 2874 | 2912 | 2915 | 2992 | 2994 | 3027 | 3031 | 3098 | 3107 | 3108 | 3114 | 3120 |
| | 3121 | 3131 | 3138 | 3141 | 3147 | 3150 | 3162 | 3163 | 3164 | 3165 | 3166 | 3167 | 3168 | 3169 | 3170 |
| | 3171 | 3172 | | | | | | | | | | | | | |
| .EQUIV | 649 | 650 | 658 | 673 | 674 | 703 | 704 | 705 | 706 | 707 | 708 | 709 | 710 | 711 | 712 |
| | 731 | 732 | 733 | 734 | 735 | 736 | 737 | 738 | 739 | 740 | | | | | |
| .EVEN | 2911 | 3137 | 3257 | | | | | | | | | | | | |
| .IF | 621 | 640 | 642 | 643 | 644 | 645 | 647 | 713 | 741 | 764 | 766 | 769 | 771 | 780 | 784 |
| | 786 | 815 | 816 | 817 | 820 | 821 | 1153 | 1155 | 1159 | 1166 | 1171 | 1173 | 1175 | 1177 | 1179 |
| | 1181 | 1182 | 1183 | 1185 | 1203 | 1223 | 1226 | 1250 | 1257 | 1260 | 1262 | 1263 | 1264 | 1277 | 1278 |
| | 1286 | 1294 | 1296 | 1298 | 1299 | 1300 | 1331 | 1337 | 1339 | 1341 | 1342 | 1343 | 1369 | 1374 | 1375 |
| | 1377 | 1378 | 1379 | 1417 | 1421 | 1423 | 1425 | 1426 | 1427 | 1455 | 1460 | 1462 | 1464 | 1465 | 1466 |
| | 1489 | 1493 | 1495 | 1497 | 1498 | 1499 | 1517 | 1522 | 1524 | 1526 | 1527 | 1528 | 1588 | 1593 | 1595 |
| | 1597 | 1598 | 1599 | 1626 | 1629 | 1631 | 1633 | 1634 | 1635 | 1683 | 1686 | 1688 | 1690 | 1691 | 1692 |
| | 1703 | 1706 | 1708 | 1710 | 1711 | 1712 | 1733 | 1736 | 1738 | 1740 | 1741 | 1742 | 1770 | 1774 | 1776 |
| | 1778 | 1779 | 1780 | 1805 | 1808 | 1810 | 1812 | 1813 | 1814 | 1837 | 1840 | 1842 | 1844 | 1845 | 1846 |
| | 1870 | 1873 | 1875 | 1877 | 1878 | 1879 | 1913 | 1917 | 1919 | 1921 | 1922 | 1923 | 1988 | 1993 | 1995 |
| | 1997 | 1998 | 1999 | 2018 | 2022 | 2024 | 2026 | 2027 | 2028 | 2074 | 2077 | 2079 | 2081 | 2082 | 2083 |
| | 2112 | 2117 | 2119 | 2121 | 2122 | 2123 | 2163 | 2167 | 2169 | 2171 | 2172 | 2173 | 2189 | 2194 | 2196 |
| | 2198 | 2199 | 2200 | 2215 | 2218 | 2220 | 2222 | 2223 | 2224 | 2267 | 2271 | 2273 | 2275 | 2276 | 2277 |
| | 2318 | 2322 | 2324 | 2326 | 2327 | 2328 | 2358 | 2362 | 2364 | 2366 | 2367 | 2368 | 2386 | 2388 | 2390 |
| | 2392 | 2393 | 2394 | 2412 | 2414 | 2416 | 2418 | 2419 | 2420 | 2440 | 2441 | 2442 | 2443 | 2444 | 2445 |

K07

MAINDEC-11-DZNWA-A MACY11 27(732) 25-SEP-76 12:05 PAGE 92
 DZNWA.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

| | | | | | | | | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 2447 | 2453 | 2456 | 2458 | 2462 | 2464 | 2470 | 2472 | 2473 | 2478 | 2482 | 2492 | 2495 | 2503 | 2517 |
| | 2520 | 2540 | 2543 | 2557 | 2560 | 2571 | 2573 | 2583 | 2586 | 2591 | 2596 | 2608 | 2610 | 2611 | 2612 |
| | 2614 | 2615 | 2616 | 2625 | 2627 | 2635 | 2636 | 2641 | 2642 | 2643 | 2645 | 2648 | 2657 | 2660 | 2664 |
| | 2666 | 2667 | 2669 | 2672 | 2679 | 2683 | 2684 | 2686 | 2688 | 2689 | 2692 | 2693 | 2721 | 2729 | 2731 |
| | 2735 | 2736 | 2779 | 2780 | 2786 | 2788 | 2809 | 2858 | 2873 | 2889 | 2914 | 2991 | 2994 | 3006 | 3030 |
| | 3097 | 3107 | 3108 | 3113 | 3120 | 3121 | 3129 | 3131 | 3135 | 3140 | 3146 | 3150 | 3154 | 3163 | 3164 |
| | 3165 | 3166 | 3167 | 3168 | 3169 | 3170 | 3171 | 3172 | | | | | | | |
| .IFF | 640 | 642 | 644 | 645 | 647 | 767 | 771 | 773 | 781 | 784 | 786 | 815 | 821 | 1155 | 1159 |
| | 1171 | 1223 | 1226 | 1250 | 1257 | 1261 | 1262 | 1263 | 1277 | 1278 | 1286 | 1294 | 1297 | 1298 | 1299 |
| | 1331 | 1337 | 1340 | 1341 | 1369 | 1374 | 1376 | 1377 | 1417 | 1421 | 1424 | 1425 | 1455 | 1460 | 1463 |
| | 1464 | 1489 | 1493 | 1496 | 1497 | 1517 | 1522 | 1525 | 1526 | 1588 | 1593 | 1596 | 1597 | 1626 | 1630 |
| | 1632 | 1633 | 1683 | 1686 | 1689 | 1690 | 1703 | 1706 | 1709 | 1710 | 1733 | 1736 | 1739 | 1740 | 1770 |
| | 1774 | 1777 | 1778 | 1805 | 1808 | 1811 | 1812 | 1837 | 1840 | 1843 | 1844 | 1870 | 1873 | 1876 | 1877 |
| | 1913 | 1917 | 1920 | 1921 | 1922 | 1988 | 1993 | 1996 | 1997 | 2018 | 2022 | 2025 | 2026 | 2074 | 2077 |
| | 2080 | 2081 | 2112 | 2117 | 2120 | 2121 | 2163 | 2167 | 2170 | 2171 | 2172 | 2189 | 2194 | 2197 | 2198 |
| | 2199 | 2215 | 2218 | 2221 | 2222 | 2223 | 2267 | 2271 | 2274 | 2275 | 2276 | 2318 | 2322 | 2325 | 2326 |
| | 2327 | 2358 | 2362 | 2365 | 2366 | 2367 | 2386 | 2388 | 2391 | 2392 | 2393 | 2412 | 2414 | 2417 | 2418 |
| | 2419 | 2441 | 2444 | 2448 | 2454 | 2457 | 2472 | 2478 | 2482 | 2492 | 2495 | 2503 | 2517 | 2520 | 2540 |
| | 2543 | 2558 | 2561 | 2571 | 2573 | 2584 | 2609 | 2612 | 2613 | 2616 | 2642 | 2646 | 2648 | 2661 | 2679 |
| | 2684 | 2687 | 2689 | 2693 | 2695 | 2700 | 2721 | 2722 | 2732 | 2763 | 2780 | 2789 | 2859 | 2873 | 2889 |
| | 2915 | 2992 | 3031 | 3098 | 3114 | 3131 | 3141 | 3147 | | | | | | | |
| .IFT | 2624 | 2667 | 2695 | 2700 | 3010 | 3026 | 3027 | | | | | | | | |
| .IFTF | 2622 | 2666 | 2689 | 2693 | 2696 | 3006 | 3010 | 3026 | | | | | | | |
| .IIF | 620 | 625 | 630 | 631 | 637 | 638 | 639 | 641 | 644 | 645 | 761 | 820 | 1172 | 1175 | 1181 |
| | 1182 | 1183 | 1185 | 1186 | 1276 | 1960 | 1971 | 2262 | 2442 | 2448 | 2449 | 2460 | 2472 | 2476 | 2587 |
| | 2588 | 2589 | 2590 | 2591 | 2592 | 2596 | 2623 | 2624 | 2639 | 2642 | 2643 | 2649 | 2650 | 2651 | 2652 |
| | 2657 | 2672 | 2679 | 2684 | 2687 | 2771 | 2780 | 2786 | 2856 | 2871 | 2896 | 2900 | 3162 | 3163 | 3164 |
| | 3165 | 3166 | 3169 | 3170 | 3171 | | | | | | | | | | |
| .IRP | 1153 | 1212 | 1260 | 1296 | 1339 | 1375 | 1423 | 1462 | 1495 | 1524 | 1595 | 1631 | 1688 | 1708 | 1738 |
| | 1776 | 1810 | 1842 | 1875 | 1919 | 1995 | 2024 | 2079 | 2119 | 2169 | 2196 | 2220 | 2273 | 2324 | 2364 |
| | 2390 | 2416 | 3001 | 3022 | 3041 | 3081 | 3101 | 3107 | 3120 | 3121 | | | | | |
| .LIST | 1 | 619 | 620 | 644 | 755 | 761 | 815 | 1153 | 1155 | 1156 | 1159 | 1160 | 1187 | 1223 | 1224 |
| | 1226 | 1227 | 1250 | 1251 | 1257 | 1258 | 1260 | 1262 | 1286 | 1287 | 1294 | 1295 | 1296 | 1298 | 1331 |
| | 1332 | 1337 | 1338 | 1339 | 1341 | 1369 | 1370 | 1374 | 1375 | 1377 | 1417 | 1418 | 1421 | 1422 | 1423 |
| | 1425 | 1455 | 1456 | 1460 | 1461 | 1462 | 1464 | 1489 | 1490 | 1493 | 1494 | 1495 | 1497 | 1517 | 1518 |
| | 1522 | 1523 | 1524 | 1526 | 1588 | 1589 | 1593 | 1594 | 1595 | 1597 | 1626 | 1627 | 1631 | 1633 | 1683 |
| | 1684 | 1686 | 1687 | 1688 | 1690 | 1703 | 1704 | 1706 | 1707 | 1708 | 1710 | 1733 | 1734 | 1736 | 1737 |
| | 1738 | 1740 | 1770 | 1771 | 1774 | 1775 | 1776 | 1778 | 1805 | 1806 | 1808 | 1809 | 1810 | 1812 | 1837 |
| | 1838 | 1840 | 1841 | 1842 | 1844 | 1870 | 1871 | 1873 | 1874 | 1875 | 1877 | 1913 | 1914 | 1917 | 1918 |
| | 1919 | 1921 | 1988 | 1989 | 1993 | 1994 | 1995 | 1997 | 2018 | 2019 | 2022 | 2023 | 2024 | 2026 | 2074 |
| | 2075 | 2077 | 2078 | 2079 | 2081 | 2112 | 2113 | 2117 | 2118 | 2119 | 2121 | 2163 | 2164 | 2167 | 2168 |
| | 2169 | 2171 | 2189 | 2190 | 2194 | 2195 | 2196 | 2198 | 2215 | 2216 | 2218 | 2219 | 2220 | 2222 | 2267 |
| | 2268 | 2271 | 2272 | 2273 | 2275 | 2318 | 2319 | 2322 | 2323 | 2324 | 2326 | 2358 | 2359 | 2362 | 2363 |
| | 2364 | 2366 | 2386 | 2387 | 2388 | 2389 | 2390 | 2392 | 2412 | 2413 | 2414 | 2415 | 2416 | 2418 | 2448 |
| | 2464 | 2478 | 2479 | 2482 | 2483 | 2492 | 2493 | 2495 | 2496 | 2503 | 2504 | 2517 | 2518 | 2520 | 2521 |
| | 2540 | 2541 | 2543 | 2544 | 2571 | 2572 | 2573 | 2574 | 2591 | 2679 | 2721 | 3154 | 3162 | 3163 | 3164 |
| | 3165 | 3166 | 3167 | 3169 | 3170 | 3171 | 3172 | | | | | | | | |
| .MACRO | 1 | 645 | 778 | | | | | | | | | | | | |
| .MCALL | 620 | 755 | 1187 | | | | | | | | | | | | |
| .MLIST | 1 | 618 | 620 | 644 | 755 | 761 | 815 | 1153 | 1155 | 1156 | 1159 | 1160 | 1187 | 1223 | 1224 |
| | 1226 | 1227 | 1250 | 1251 | 1257 | 1258 | 1260 | 1262 | 1286 | 1287 | 1294 | 1295 | 1296 | 1298 | 1331 |
| | 1332 | 1337 | 1338 | 1339 | 1341 | 1369 | 1370 | 1374 | 1375 | 1377 | 1417 | 1418 | 1421 | 1422 | 1423 |
| | 1425 | 1455 | 1456 | 1460 | 1461 | 1462 | 1464 | 1489 | 1490 | 1493 | 1494 | 1495 | 1497 | 1517 | 1518 |
| | 1522 | 1523 | 1524 | 1526 | 1588 | 1589 | 1593 | 1594 | 1595 | 1597 | 1626 | 1627 | 1631 | 1633 | 1683 |
| | 1684 | 1686 | 1687 | 1688 | 1690 | 1703 | 1704 | 1706 | 1707 | 1708 | 1710 | 1733 | 1734 | 1736 | 1737 |
| | 1738 | 1740 | 1770 | 1771 | 1774 | 1775 | 1776 | 1778 | 1805 | 1806 | 1808 | 1809 | 1810 | 1812 | 1837 |

| | | | | | | | | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 1838 | 1840 | 1841 | 1842 | 1644 | 1870 | 1871 | 1873 | 1874 | 1875 | 1877 | 1913 | 1914 | 1917 | 1918 |
| | 1919 | 1921 | 1988 | 1989 | 1993 | 1994 | 1995 | 1997 | 2018 | 2019 | 2022 | 2023 | 2024 | 2026 | 2074 |
| | 2075 | 2077 | 2078 | 2079 | 2081 | 2112 | 2113 | 2117 | 2118 | 2119 | 2121 | 2163 | 2164 | 2167 | 2168 |
| | 2169 | 2171 | 2189 | 2190 | 2194 | 2195 | 2196 | 2198 | 2215 | 2216 | 2218 | 2219 | 2220 | 2222 | 2267 |
| | 2268 | 2271 | 2272 | 2273 | 2275 | 2318 | 2319 | 2322 | 2323 | 2324 | 2326 | 2358 | 2359 | 2362 | 2363 |
| | 2364 | 2366 | 2386 | 2387 | 2388 | 2389 | 2390 | 2392 | 2412 | 2413 | 2414 | 2415 | 2416 | 2418 | 2448 |
| | 2464 | 2478 | 2479 | 2482 | 2483 | 2492 | 2493 | 2495 | 2496 | 2503 | 2504 | 2517 | 2518 | 2520 | 2521 |
| | 2540 | 2541 | 2543 | 2544 | 2571 | 2572 | 2573 | 2574 | 2591 | 2679 | 2721 | 3154 | 3162 | 3163 | 3164 |
| | 3165 | 3166 | 3167 | 3169 | 3170 | 3171 | 3172 | | | | | | | | |
| .PAGE | 778 | 821 | | | | | | | | | | | | | |
| .REM | 1 | | | | | | | | | | | | | | |
| .REPT | 761 | 1155 | 1159 | 1223 | 1226 | 1250 | 1257 | 1286 | 1294 | 1331 | 1337 | 1369 | 1374 | 1417 | 1421 |
| | 1455 | 1460 | 1489 | 1493 | 1517 | 1522 | 1588 | 1593 | 1626 | 1683 | 1686 | 1703 | 1706 | 1733 | 1736 |
| | 1770 | 1774 | 1805 | 1808 | 1837 | 1840 | 1870 | 1873 | 1913 | 1917 | 1988 | 1993 | 2018 | 2022 | 2074 |
| | 2077 | 2112 | 2117 | 2163 | 2167 | 2189 | 2194 | 2215 | 2218 | 2267 | 2271 | 2318 | 2322 | 2358 | 2362 |
| .SBTTL | 2386 | 2388 | 2412 | 2414 | 2478 | 2482 | 2492 | 2495 | 2503 | 2517 | 2520 | 2540 | 2543 | 2571 | 2573 |
| | 633 | 645 | 755 | 764 | 778 | 821 | 1165 | 2438 | 2581 | 2643 | 2684 | 2786 | 2856 | 2912 | 2989 |
| | 3028 | 3095 | 3138 | 3154 | | | | | | | | | | | |
| .TITLE | 620 | | | | | | | | | | | | | | |
| .WORD | 761 | 762 | 763 | 772 | 786 | 789 | 790 | 791 | 792 | 795 | 796 | 797 | 798 | 799 | 800 |
| | 801 | 804 | 805 | 806 | 1062 | 1126 | 1127 | 1128 | 1129 | 1130 | 1131 | 1132 | 1133 | 1134 | 1135 |
| | 1136 | 1137 | 1138 | 1139 | 1140 | 1141 | 1142 | 1143 | 1144 | 1145 | 1148 | 1324 | 1325 | 1326 | 1327 |
| | 2453 | 2456 | 2471 | 2538 | 2853 | 2882 | 2887 | 2988 | 3026 | 3130 | | | | | |

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* ,DZNWA.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZNWA.SRC
RUN-TIME: 40 55 7 SECONDS
RUN-TIME RATIO: 310/103=2.9
CORE USED: 34K (67 PAGES)

