

# PDP11

WATCHDOG TIMER  
MD-11-DBQAA-A

EP-DBQAA-A-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames, each displaying technical data. The data is organized into columns and rows, with some frames containing headers and footers. The content includes various alphanumeric strings, possibly representing memory addresses, register values, or diagnostic codes. The frames are arranged in a regular grid pattern, typical of microfiche storage.



B01

MAIN MACY11 27(732) 15-OCT-76 11:29 PAGE 2  
DBQAA P11

.REM 2

DBQAA P11

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DBQAA-A-D
PRODUCT NAME:	WATCH DOG DIAGNOSTIC
DATE:	
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GEORGE BAISLEY

COPYRIGHT      MARCH 1973  
 DIGITAL EQUIPMENT CORPORATION  
 MAYNARD, MASSACHUSSETTS 01754

3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80

1. ABSTRACT

THE FUNCTION OF THE WATCHDOG DIAGNOSTICS IS TO VERIFY THAT THE WATCH DOG TIMER PERFORMS AS SPECIFIED BY THE PRINT SET. THE WATCH DOG PACKAGE CONSISTS OF ONE TAPE. THE WATCH DOG TIMER HARDWARE IS COMPOSED OF TWO INDEPENDENT FUNCTIONAL PARTS: A. WATCH DOG TIMER AND B. WATCH DOG LINE CLOCK EACH HAVE THEIR OWN VECTOR AND DEVICE ADDRESSES. THE LINE CLOCK PORTION OF THE DIAGNOSTICS REQUIRES THE OPERATOR TO MONITOR IT'S OPERATION WITH A CLOCK CAPABLE OF MEASURING TIME IN SECONDS.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11/40 ONLY WITH WATCH DOG TIMER

PDP-11/20 MAY BE USED, HOWEVER IT WILL NOT EXERCISE THE RESTART LOGIC PORTION AND THEREFORE WILL FAIL THOSE FUNCTIONS.

2.2 STORAGE

2.2.1. PROGRAM STORAGE

THIS PROGRAM LOADS IN 4K OF MEMORY

3. LOADING PROCEDURE

3.1 METHOD

PROGRAM IS IN ABSOLUTE FORMAT AND IS LOADED USING THE ABSOLUTE LOADER

ABSOLUTE LOADER START ADDRESS #500

MEMORY SIZE	*
4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

3.1.1. LOAD ADDRESS OF ABS LOADER INTO SWITCHES

3.1.2. DEPRESS "LOAD ADDRESS" KEY ON CONSOLE

3.1.3. DEPRESS "START" KEY ON CONSOLE

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

- 4. STARTING PROCEDURE
  - A. SET SWITCH REGISTER TO 000200
  - B. DEPRESS "LOAD ADDRESS" KEY
  - C. DEPRESS START

THE PROGRAM WILL JUMP TO THE DIAGNOSTIC MONITOR AND TYPE OUT THE OPERATING INSTRUCTIONS. THIS IS ONCE ONLY CODE. TO RETYPE THE OPERATING INSTRUCTION THE OPERATOR MAY EITHER RELOAD THE PROGRAM OR LOAD THE ADDRESS "MONITOR" IN THE SWITCH REGISTER AND DEPRESS START.

4.1 CONTROL SWITCH SETTINGS

- SR15 HALT ON ERROR
- SR14 SCOPE LOOP (SCOPE ON TEST OR ERROR)
- SR13 INHIBIT ERROR PRINTOUT
- SR12 SHORT ERROR REPORT
- SR11 INHIBIT ITERATIONS

4.2 STARTING ADDRESSES

ADDRESSES	COMMENT
000200	NORMAL START
	WITH 200 LEFT IN THE SWITCHES THE PROGRAM TYPES OUT FULL INSTRUCTIONS ONCE AND ABBREVIATED INSTRUCTIONS THEREAFTER. WITH THE SWITCHES ZERO THE PROGRAM SETUP EITHER THE DEFAULT OR PREVIOUSLY SELECTED PARAMETERS AND IMMEDIATELY ASKS FOR DYNAMIC SWITCH SETTINGS.
MONITOR	RELOAD TAPE FOR RETYPING OF INSTRUCTIONS
000042	IF THIS LOCATION IS NONZERO THE PROGRAM ASSUMES IT IS RUNNING UNDER ACT11 OR DDP AND USES THE DEFAULT PARAMETERS

125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177

## 5. OPERATING PROCEDURE

STARTING FROM 200 CAUSES THE FOLLOWING TYPE OUT.

MAINDEC-11-DBGAA-A-D

DYNAMIC SWITCH SETTINGS

SR15 HALT ON ERROR  
SR14 SCOPE LOOP  
SR13 INHIBIT ERROR PRINTOUT  
SR12 SHORT ERROR REPORT  
SR11 INHIBIT ITERATIONSTYPE: <D>, FOR DEFAULT PARAMETERS  
<P>, FOR PREVIOUS PARAMETERS  
<S>, FOR SELECT PARAMETERS  
<N>, FOR START WITH THIS TEST NUMBER

D,P,S,N?

IN RESPONSE TO THIS LAST QUESTION THE OPERATOR IS REQUIRED TO TYPE ONE OF THE LETTERS IN THE STRING. ON THE FIRST TIME THRU THIS DIAGNOSTIC TYPE "D" FOR DEFAULT TO SET UP PARAMETERS, OR TYPE "S" TO SELECT PARAMETERS.

TYPE "P" OR "N" AFTER THE PARAMETERS HAVE BEEN SELECTED FROM PREVIOUS RUNS.

IF ANY CHARACTER OTHER THAN ONE IN THE STRING IS TYPED THE MONITOR WILL REJECT THE CHARACTER AND RETYPE THE STRING.

IF, IN RESPONSE TO THE STRING, THE OPERATOR TYPES AN "S" THE SELECTION SEQUENCE IS ENTERED AND THE FOLLOWING DIALOGUE TAKES PLACE.

NOTE: THESE ARE THE DEFAULT PARAMETERS: TYPING &lt;CR&gt; IS EQUIVALENT TO TYPING THE DEFAULT PARAMETER.

TEST NUMBER: 1  
BASE ADDRESS: 172400  
VECTOR ADDRESS: 300  
WD PRIORITY LEVEL: 6TIMEOUT VALUE IN MILLISECS + TOLERANCE: 140  
TRIGGER VALUE IN MILLISECS (5% OF TIMEOUT VALUE): 7CLOCK FREQ IN HZ: 1000  
RESTART ADDRESS: 173776  
SET SWITCHES

AT ANY TIME DURING THE 'S' ELECTION SEQUENCE A CONTROL C MAY BE TYPED AND THE MONITOR WILL ASK AGAIN "D,P,S,N?"

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

## 5.1 "TEST NUMBER: 1"

HERE THE MONITOR IS ASKING FOR THE NUMBER OF THE FIRST TEST IN THE CHAINING SEQUENCE. THE DEFAULT ANSWER IS (1) ONE, THE FIRST TEST IN THE CHAIN. THE OPERATOR MAY TYPE ANY TEST NUMBER AS SPECIFIED IN THE LISTING; HOWEVER, AT THIS WRITING THERE IS NO CHECK TO SEE IF THE OPERATOR SELECTED A NONEXISTANT TEST NUMBER. HENCEFORTH THE NUMBER TYPED AFTER THE COLON IS THE DEFAULT PARAMETER UNTIL CHANGED.... TYPING A <CR> IS EQUIVALENT TO TYPING THIS DEFAULT PARAMETER.

"BASE ADDRESS: 172400"

THIS IS THE BASE ADDRESS FOR THE WATCH DOG TIMER AND ASSOCIATED LINE CLOCK. BY TYPING THE CORRECT BASE ADDRESS; CLKCSR, WDCSR, AND WDDAT ARE GENERATED AUTOMATICALLY. EG. 172400, 172404, 172406 RESPECTIVELY

TYPING <CR> WILL GENERATE THE DEFAULT PARAMETERS. 172400, 172404, 172406

"VECTOR ADDRESS: 300"

THE WATCH DOG TIMER IS CUT TO INTERRUPT TO ADDRESS 300 AT THE FACTORY. BY TYPING THE CORRECT VECTOR ADDRESS; CLKIV, CLKIS, WDIV, AND WDIS ARE GENERATED AUTOMATICALLY EG. 300, 302, 304, 306 RESPECTIVELY

TYPING <CR> WILL GENERATE THE DEFAULT PARAMETERS 300, 302, 304, 306

"WD PRIORITY LEVEL: 6"

THIS IS THE DEVICE INTERRUPT PRIORITY LEVEL - THE W.D.T. IS FACTORY WIRED TO LEVEL 6. BY TYPING THE WRONG PRIORITY LEVEL THE DIAGNOSTIC WILL RESPOND WITH ERRORS IN THE PROGRAM ACCORDINGLY.

TYPING <CR> WILL SET UP THE DEFAULT VALUE 6.

"TIMEOUT VALUE IN MILLISECS + TOLERANCE : 140"

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

TIMEOUT (OR TIMOUT) IS THE WATCH DOG'S TIME OUT VALUE. THAT IS, UNLESS IT IS "UPDATED" WITHIN THE PRECUT VALUE THE WATCH DOG RELAY WILL FALL CAUSING RESTART, ETC.

BY TYPING THE CORRECT TIMOUT VALUE IN MILLISECS (DECIMAL NUMBERS ONLY, NO FRACTIONS OR DECIMAL POINT), THE DIAGNOSTIC WILL PASS THE TESTS. BY REDUCING THIS NUMBER AT THE MONITOR LEVEL (TYPING "S" ELECT PARAMETERS) AND ALTERNATELY RUNNING THE PROGRAM THE ACTUAL PRECUT VALUE OF TIMOUT CAN BE ASCERTAINED.... THE PROGRAM WILL FAIL BY SPECIFYING THAT A TOO LOW VALUE OF TIMOUT WAS TYPED IN. BY REPEATEDLY RUNNING THE PROGRAM AND RESELECTING THE TIMOUT VALUE, IT CAN BE DETERMINED TO AN ACCURACY OF + 1 MSEC.

TYPING <CR> WILL SET UP THE DEFAULT VALUE 140 (MILLISECONDS IN DECIMAL NUMBERS)

NOTE1: TYPE WHOLE DECIMAL NUMBERS ONLY, FRACTIONS WILL CAUSE ERRORS, THE LOWEST ACCEPTABLE NUMBER IS 1.

NOTE2: THE HIGHEST VALUE THAT WILL BE ACCEPTED IS 10000 MILLISECONDS (10 SECS) TYPING A NUMBER HIGHER THAN 10000 WILL CAUSE ERRORS WITHIN THE PROGRAM (DO NOT USE COMMAS).

"TRIGER VALUE IN MILLISECS (5% OF TIMOUT VALUE): 7"

TRIGGER (OR TRIGER) IS THE WATCH DOG'S TOO SOON (OR TOOSN) VALUE. THAT IS, IF THE WATCH DOG IS "UPDATED" IN LESS TIME THAN 5% OF TIMOUT VALUE THE WATCH DOG WILL GENERATE THE TOOSN BIT AND CAUSE INTERRUPT (IF INTERRUPT ENABLE IS SET).

BY TYPING THE CORRECT TRIGER VALUE IN MILLISECS (DECIMAL NUMBERS ONLY, NO FRACTIONS, OR DECIMAL POINT), THE DIAGNOSTIC WILL PASS THE TESTS. BY REDUCING THIS NUMBER AT THE MONITOR LEVEL (TYPING "S" ELECT PARAMETERS) AND ALTERNATELY RUNNING THE PROGRAM, THE ACTUAL PRECUT VALUE OF TRIGER CAN BE ASCERTAINED.... THE PROGRAM WILL FAIL BY SPECIFYING THAT A TOO LOW VALUE OF TRIGER WAS TYPED IN. BY REPEATEDLY RUNNING THE PROGRAM AND RESELECTING THE TRIGER VALUE, IT CAN BE DETERMINED TO AN ACCURACY OF + 1 MSEC.

TYPING <CR> WILL SET UP THE DEFAULT VALUE 7 (MILLISECONDS IN DECIMAL NUMBERS)

NOTE1: TYPE WHOLE DECIMAL NUMBERS ONLY, FRACTIONS WILL CAUSE ERRORS; THE LOWEST ACCEPTABLE NUMBER IS 1.

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

NOTE2: THE HIGHEST VALUE THAT WILL BE ACCEPTED IS 10000 MILL-ISECONDS (10 SECS). TYPING A NUMBER HIGHER THAN 10000 WILL CAUSE ERRORS WITHIN THE PROGRAM (DO NOT USE COMMAS). THIS NUMBER SHOULD NEVER HAVE TO BE USED AS TRIGER VALUE WILL NOT EXCEED 5% OF TIMEOUT VALUE.

"CLOCK FREQ IN HZ: 1000"

THIS IS THE WATCH DOG'S LINE CLOCK FREQUENCY PRECUT AT THE FACTORY. OTHER ACCEPTABLE FREQUENCIES ARE 50, 60, 100, AND 200 HZ. TYPING OTHER THAN SPECIFIED FREQUENCIES WILL GENERATE AN ERROR WITHIN THE PROGRAM.

TYPING <CR> WILL SET UP THE DEFAULT VALUE 1000

"RESTART ADDRESS: 173776"

THIS NUMBER IS THE RESTART ADDRESS..... IT IS NECESSARY THAT BEFORE RUNNING THE DIAGNOSTIC THAT THIS NUMBER BE STRAPPED OR SWITCHED IN THE WATCH DOG TIMER, DO NOT USE ANY VALID R.O.M. STARTING ADDRESS.....

SHOULD THE ABOVE NUMBER BE IN CONFLICT WITH ANY VALID R.O.M. ADDRESS, CHOOSE ANY NUMBER FROM 173000 TO 173776 AND STRAP THE WATCH DOG TIMER ACCORDINGLY.

"SET SWITCHES"

HERE THE MONITOR ASKS FOR THE CONSOLE SWITCH SETTINGS

- SR15 HALT ON ERROR
- SR14 SCOPE LOOP
- SR13 INHIBIT ERROR PRINTOUT
- SR12 SHORT ERROR REPORT
- SR11 INHIBIT ITERATIONS

LOAD THE SWITCH REGISTER WITH THE APPROPRIATE FUNCTION AND TYPE <CR>

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. SET SR15 - HALT ON ERROR WHEN AN ERROR HALT OCCURS.
2. CLEAR SR15
3. SET SR14 AND DEPRESS "CONTINUE" ON CONSOLE. PROGRAM WILL SCOPE ON ERROR.
4. IF ERROR IS REPETITIVE SET SR13 AND SCOPE ERROR



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

5. SET SR11 IF THE TEST IS FAILING ON A PARTICULAR PASS I.E.  
2ND AND NOT THE 1ST

THE ERROR PC SHOULD BRING THE OPERATOR TO A POINT IN THE LISTING  
WHERE THE ERROR IS DOCUMENTED. THEN USING THE PRINTS AND THE  
FLOWS THE ERROR CAN BE TRACED TO ITS SOURCE.

AT ANY TIME DURING THE INITIALIZATION OR TESTING THE OPERATOR  
CAN TYPE CONTROL C AND CONTROL WILL BE RETURNED TO THE MONITOR.  
SOME TESTS ARE 5 TO 10 SECONDS IN DURATION SO THE RESPONSE TO  
CONTROL C WILL NOT BE INSTANTANEOUS.

THE RESTART ADDRESS IS 200. IF THIS ADDRESS IS LEFT IN THE CONSOLE  
SWITCHES WHEN "START" IS PRESSED, THE MONITOR WILL TYPE OUT  
P, P, S, N?. IF THE SWITCHES ARE ZEROED THE TYPE OUT WILL BE  
"SET SWITCHES".

## 6. ERRORS

TYPICALLY ERROR REPORTS TAKE THE FOLLOWING FORMAT.

ERROR PC 002776

THIS INDICATES THAT WHILE EXECUTING THE TESTS AN ERROR STATE WAS  
DETECTED AND IS DOCUMENTED AT PROGRAM COUNT 002776 AND TEST 3

A TYPICAL APPROACH MIGHT BE; TYPE CONTROL C, RESULTS

D.P.S.N? N  
TEST NUMBER: 3  
SET SWITCHES

IN RESPONSE TO SWITCHES SET THE FOLLOWING

SR15=0 HALT ON ERROR  
SR14=1 SCOPE  
TYPE <CR>

IF THE ERROR IS REPEATABLE SET SR13 INHIBIT PRINT AND START  
TROUBLE SHOOTING. REFER TO LISTING FOR THE COMMENTS ASSOCIATED  
WITH THE ERROR.

\*\*\*\*\*

ADDED NOTE: MANY ERRORS THAT MAY BE ENCOUNTERED WILL NOT BE THERE  
WHEN EXAMINING THE CONTENTS OF THE SWITCH REGISTER. THIS IS  
PRIMARILY DUE TO THE TIMING CHARACTERISTICS WITHIN THE WATCH DOG  
TIMER. IN MANY CASES WHEN TIMING PLAYS AN IMPORTANT ROLE IN THE  
DIAGNOSTIC ROUTINE, THE DATA IS STORED AT A TEMPORARY TAG  
LOCATION PRIOR TO HALTING ON ERROR. THEREFORE, BACK-UP INTO THE  
DIAGNOSTICS TO FIND WHERE THIS DATA IS TEMPORARILY HELD; LOAD  
IT'S PC INTO THE SWITCH REGISTER AND 'EXAMINE'. GOOD LUCK!

\*\*\*\*\*

### 6.2 ERROR RECOVERY

PROGRAM CAN BE RESTARTED AT ANY TIME BY LOAD ADDRESS 200 AND

J01

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 10  
DBGAAA.P11

374

DEPRESS START

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  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

- A. DISCONNECT THE USERS CABLE AND INSTALL THE JUMPER PLUG THAT WAS SUPPLIED WITH THE WATCH DOG TIMER.
- B. STRAP THE RESTART ADDRESS TO AN UNUSED R.O.M. ADDRESS PREFERABLY 173776. SHOULD THIS ADDRESS BE USED, CHOOSE ANY OTHER ADDRESS BETWEEN 173000 TO 173776 AND LOAD THIS WHEN "S" ELECTING PARAMETERS.
- C. SHOULD THE JUMPER PLUG BE UNAVAILABLE; JUMPER THE FOLLOWING:

PIN 8 TO PIN 2  
 PIN 5 TO PIN 7  
 PIN 6 TO PIN 3  
 PIN 10 TO PIN 3  
 PIN 11 TO PIN 1, THRU A SWITCH (RESTART SWITCH)  
 PIN 9 TO PIN 1  
 PIN 17 TO PIN 16  
 PIN 15 TO PIN 14

ALL OF THESE JUMPERS MUST BE IN PRIOR TO RUNNING ANY OF THE FEED BACK LOOP TESTS. THEY ARE NOT REQUIRED IF ONLY THE LINE CLOCK PORTION OF THE WATCH DOG IS TO BE TESTED. SEE LISTINGS FOR DETAILED INFORMATION.

8. MISCELLANEOUS

- A. DISCONNECT USERS CABLE AND INSERT JUMPER PLUG - SEE SECTION 7.1
- B. IT IS IMPORTANT TO RUN THE LINE CLOCK PORTION BEFORE ATTEMPTING TO RUN THE WATCH DOG PORTION; IN PARTICULAR THE MNCLK TEST AS ALL WATCH DOG TIMING IS BASED UPON THE PRESENCE OF THE MAINTENANCE CLOCK (MNCLK).
- C. LINE CLOCK ACCURACY TESTS AT BEST ARE CLOSE APPROXIMATIONS, PRECISION MEASUREMENT IS BEYOND THE SCOPE OF THIS DIAGNOSTIC.

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

9. PROGRAM DESCRIPTION

THE PROGRAM CONSISTS OF A SERIES OF SMALL TESTS THAT CHECK EACH OF THE INDIVIDUAL FUNCTIONS AND CHARACTERISTICS OF THE WATCH DOG TIMER AND ASSOCIATED LINE CLOCK.

TEST NAME	DESCRIPTION
TEST 1 (LINE CLOCK)	TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
TEST 2 (LINE CLOCK)	TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
TEST 3 (LINE CLOCK)	TEST THAT CLOCK INTERRUPT PUSHES CONDITION CODES AND PROGRAM COUNTER ONTO STACK
TEST 4 (LINE CLOCK)	TEST THAT RESET SETS CLOCK FLAG (TICK)
TEST 5 (LINE CLOCK)	TEST LINE CLOCK REPEATABILITY TO MAKE SURE THAT OVER TWO EQUAL PERIODS OF TIME THE LINE CLOCK PUTS OUT THE SAME NUMBER OF PULSES.
TEST 6 (LINE CLOCK)	TEST LINE CLOCK REPEATABILITY WITH INTERRUPT
TEST 7 (LINE CLOCK)	LINE CLOCK ACCURACY TEST (CLOCK CAPABLE OF MEASURING TIME IN SECONDS IS REQUIPED)

449	TEST 10	MAINTENANCE CLOCK BIT TEST (MNCLK)
450	(LINE CLOCK/WD TIMER)	
451	TEST 11	MNCLK ACCURACY TEST
452	(LINE CLOCK/WD TIMER)	(CLOCK CAPABLE OF MEASURING TIME
453		IN SECONDS IS REQUIRED)
454		
455	TEST 12	TEST THAT CHECKS BITS: READ/WRITE
456	(WDT)	ABILITY OF WD DATA (WDDAT)
457		
458	TEST 13	TEST THAT CHECKS TOO SOON (TOOSN)
459	(WDT)	BIT; READ/SET ABILITY OF WDCSR
460		
461	TEST 14	TEST THAT CHECKS BITS WD INTERRUPT
462	(WDT)	ENABLE (WDIEN), EXTERNAL REQUEST
463		(EXTREQ), AND WD ENABLE (WDENAB);
464		READ/WRITE ABILITY OF WDCSR
465		
466	TEST 15	TEST THAT WATCH DOG INTERRUPTS TO
467	(WDT)	CORRECT VECTOR ADDRESS AND
468		TOOSN BIT IS SET BY "UPDATING" TOO
469		FAST (TOO SOON)
470		
471	TEST 16	TEST THAT CHECKS THE UPDATE SEQUENCE
472	(WDT)	IN REVERSE DOES NOT CAUSE AN ACT-
473		UAL 'UPDATE'
474		
475	TEST 17	TEST THAT CHECKS WD RELAY IS PICKED
476	(WDT)	AND DROPS IN TIME SPECIFIED BY
477		TIMOUT. THIS TEST CAN BE USED
478		TO DETERMINE ACTUAL TIME OUT VALUE
479		+ OR - 1 MSEC. SEE SECTION 5.1
480		
481		NOTE: IT IS IMPORTANT THAT TIMEOUT
482		BE SELECTED TO PASS THIS TEST WITH
483		A VALUE JUST OVER THE PRECUT VALUE
484		OF "TIMEOUT".
485		
486		

487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504

TEST 20  
(WDT)

TEST THAT CHECKS TOOSN BIT NOT  
SET IN TIME SPECIFIED BY TRIGER.  
THIS TEST CAN BE USED TO DETERMINE  
ACTUAL TRIGGER VALUE + OR - 1 MSEC.  
SEE SECTION 5.1

NOTE: IT IS IMPORTANT THAT TRIGER  
BE SELECTED TO PASS THIS TEST WITH  
A VALUE JUST OVER THE PRECUT VALUE  
OF "TRIGGER".

TEST 21  
(FEEDBACK LOOP)

TEST THAT CHECKS REMOTE SENSE LOGIC.  
START LOGIC AND RESTART LOGIC.  
IT IS IMPORTANT THAT THE DEVICE  
IS OPERATING ON AN 11/40 FOR THIS  
TEST, ALSO RESTART ADDRESS SHALL  
BE DETERMINED PRIOR TO RUNNING  
THIS TEST.

505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600

TEST 22  
(FEED BACK LOOP)

TEST THAT EXTREQ BIT SETS AND THAT INTERRUPT IS GENERATED WHEN RELAY DROPS (CLEARS)

TEST 23  
(FEED BACK LOOP)

TEST THAT CHECKS THE MD TRIGGER O.S. TRANSISTOR. DETERMINES THAT THE TRANSISTOR IS OPERATING CORRECTLY AND NOT SHORTED OR OPEN.

TEST 24  
(END OF FEEDBACK LOOP)

THIS IS NOT A TEST. TYPES "DO RESTART TEST, CLOSE SWITCH" THE PROGRAM HAS COMPLETED THE FIRST SECTION OF TESTS - THE PROGRAM WILL NOW PROCEED TO REPEAT THE PRECEDING TESTS. TYPE IC TO RETURN TO MONITOR, TO DO THE RESTART TEST. UNLESS IC IS TYPED, THE PROGRAM WILL TEST THE FIRST SECTION INDEFINITELY.

TEST 25  
(RESTART TEST)

TEST THAT CORRECT RESTART ADDRESS IS GENERATED AND RESTART LOGIC IS FUNCTIONING

NOTE1: HALT CPU BEFORE PROCEEDING INTO THIS TEST  
NOTE2: CLOSE SWITCH (RESTART SWITCH ON JUMPER PLUG)  
NOTE3: TYPE IC TO RETURN TO MONITOR  
NOTE4: TYPE N, CR, 25, CR CR (CONTINUE AS NORMAL)

TEST 26  
(END RESTART TEST)

THIS NOT A TEST. TYPES "END TEST" THE PROGRAM HAS COMPLETED THE SECOND SECTION TYPE IC TO STOP PROGRAM. UNLESS IC IS TYPED, THE PROGRAM WILL TEST THIS SECTION INDEFINITELY.

- 10. LISTING
- 11. FLOW CHART (WDFLOW.LST)
- 12. TIMING DIAGRAM (WDTIME.LST)

560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615

2

000000

. = 0

.REM \*

MAINDEC-11-DBQAA-A-D  
COPYRIGHT 1973 DIGITAL EQUIPMENT CORP.  
146 MAIN ST. MAYNARD, MA. 01754  
MAINTAINER: DIAGNOSTICS  
AUTHOR: G. BAISLEY

\*

000000 000000  
000002 000000  
000004 000006  
000006 000000  
000010 000012  
000012 000000  
000014 000016  
000016 000000

HALT  
HALT  
. +2  
HALT  
. +2  
HALT  
. +2  
HALT

: TRAP HERE IF ADDRESS  
: LINES DROP  
: TRAP PC POINTS TO HALT  
: TRAP PC POINTS TO HALT  
: TRAP PC POINTS TO HALT







728	000360	000362	.+2	; TRAP PC POINTS TO HALT
729	000362	000000	HALT	
730	000364	000366	.+2	; TRAP PC POINTS TO HALT
731	000366	000000	HALT	
732	000370	000372	.+2	; TRAP PC POINTS TO HALT
733	000372	000000	HALT	
734	000374	000376	.+2	; TRAP PC POINTS TO HALT
735	000376	000000	HALT	
736	000400	000402	.+2	; TRAP PC POINTS TO HALT
737	000402	000000	HALT	
738				
739		104400	SCOPE=TRAP	;SCOPE LOOP TRAP
740				
741		100000	BIT15=100000	
742		040000	BIT14=40000	
743		020000	BIT13=20000	
744		010000	BIT12=10000	
745		004000	BIT11=4000	
746		002000	BIT10=2000	
747		001000	BIT9=1000	
748		000400	BIT8=400	
749		000200	BIT7=200	
750		000100	BIT6=100	
751		000040	BIT5=40	
752		000020	BIT4=20	
753		000010	BIT3=10	
754		000004	BIT2=4	
755		000002	BIT1=2	
756		000001	BIT0=1	
757		000000	HERE=0	
758			;REGISTER DEFINITIONS	
759				
760		000000	R0=%0	
761		000001	R1=%1	
762		000002	R2=%2	
763		000003	R3=%3	
764		000004	R4=%4	
765		000005	R5=%5	
766		000005	TTY=%5	
767		000006	R6=%6	
768		000006	SP=%6	;STACK POINTER
769		000007	PC=%7	;PROGRAM COUNTER
770				
771		000004	TYPE=IOT	
772		000240	NOP=240	
773		177776	PS=177776	;PROCESSOR STATUS
774		177570	SWR=177570	
775		177570	SR=177570	;SWITCH REGISTER
776			;PROCESSOR PRIORITY LEVELS	
777		000000	LEVEL0=000	
778		000040	LEVEL1=040	
779		000100	LEVEL2=100	
780		000140	LEVEL3=140	
781		000200	LEVEL4=200	
782		000240	LEVEL5=240	
783		000300	LEVEL6=300	

```

784          000340          LEVEL7=340
785
786
787          :TTY ADDRESSES
788 000404 177560          TKS:177560
789 000406 177562          TKB:177562
790 000410 177564          TPS:177564
791 000412 177566          TPB:177566
792
793          :MISC DEFINITIONS
794          000015          CR=15
795          000012          LF=12
796
797          ;EMT DEFINITIONS
798
799 000414 104000          ERROR          ;TRAPS TO T.ERROR
800 000416 104001          SAVRG          ;TRAPS TO T.SAVRG
801 000420 104002          RSTRG          ;TRAPS TO T.RSTRG
802 000422 104003          ACCEPTO          ;TRAPS TO T.ACCEPTO
803 000424 104004          KEY.TO.RO          ;TRAPS TO T.KEY.TO.RO
804
805
806          ;TRAP INITIALIZATION
807
808          .=20
809 000020 013150 000340          .IOT,LEVEL7          ;TTY OUTPUT TRAP,LEVEL 7
810
811          .=24
812 000024 013552 000340          PFAIL,LEVEL7          ;POWER FAIL TRAP
813
814          .=30
815 000030 012322 000340          EMTDECODER,LEVEL7          ;EMT DECODER TRAP,LEVEL 7
816
817          .=34
818 000034 012536 000340          SCOPEC,LEVEL7          ;SCOPE TRAP
819 000200          .=200
820
821 000200 000137 001100          START: JMP          @#BEGIN          ;GO TO BEGINNING OF PROGRAM
822
823          .=1100
824
825 001100 012706 001100          BEGIN: MOV          @#BEGIN,SP          ;SET UP STACK POINTER
826 001104 012767 000340 176664          MOV          @#LEVEL7,PS          ;PRIORITY LEVEL 7
827 001112 005737 000042          TST          @#42          ;ACT11
828 001116 001404          BEQ          BGNO          ;BR IF NO
829 001120 004767 006214          JSR          PC,DEFAULT          ;INSERT DEFAULT PARAMETERS
830 001124 000137 007310          JMP          @#MON13
831 001130 005327 000001          BGNO: DEC          @1          ;ONCE ONLY
832 001134 001002          BNE          BGN1
833 001136 000137 006346          JMP          @#MONITOR
834 001142 032767 000200 176420          BGN1: BIT          @200,SR          ;TEST FOR FAST START
835 001150 001402          BEQ          BGN2          ;BRANCH IF FAST START
836 001152 000137 006404          JMP          @#MON1.0
837 001156 012706 001100          BGN2: MOV          @#BEGIN,SP
838 001162 012767 000340 176606          MOV          @#LEVEL7,PS
839 001170 000137 007252          JMP          @#MON10          ;USE PREVIOUS PARAMETERS
  
```

```

840
841
842
843      000200
844      000100
845
846      000200
847      000100
848      000040
849      000020
850      000010
851      000004
852      000002
853      000001
854
855      000010
856      000004
857      000002
858      000001
859
860 001174 000000
861 001176 000000
862 001200 000000
863 001202 000000
864
865 001204 000000
866 001206 000000
867 001210 000000
868 001212 000000
869
870 001214 000300
871 001216 000302
872 001220 000304
873 001222 000306
874 001224 172400
875 001226 172402
876 001230 172404
877 001232 172406
878
879 001234 000214
880 001236 000007
881 001240 001750
882 001242 173776
883 001244 176030
884 001246 166170
885 001250 154360
886
887
888
889 001252 000030
890 001254 000005
891 001256 000300
892 001260 000000
893 001262 003720
894 001264 000000
895 001266 000000
    
```

```

;DEFINITIONS
:CLOCK CSR XXX0
TICK=BIT7
CLKEN=BIT6
;WATCH DOG CSR XXX4
FLAG=BIT7
WDIEN=BIT6
TOOSN=BIT5
REMSEN=BIT4
EXTREQ=BIT3
WDENAB=BIT2
WDRLY=BIT1
X=BIT0
;WATCH DOG DATA XXX6
MNCLK=BIT3
WD2=BIT2
WD1=BIT1
X=BIT0
;WATCH DOG VECTOR AND DEVICE ADDRESSES
CLKIV:0
CLKIS:0
WDIV:0
WDIS:0
;WATCH DOG ADDRESSES
CLKCSR:0
VOID:0
WDCSR:0
WDDAT:0
;DEFAULT VECTOR AND DEVICE ADDRESS
DFLTIV: 300 ;CLKIV
        302 ;CLKIS
        304 ;WDIV
        306 ;WDIS
DFLTAD: 172400 ;CLKCSR
        172402 ;VOID
        172404 ;WDCSR
        172406 ;WDDAT
;DEFAULT MISC. PARAMETERS
DFLMSC: 140. ;TIMOUT
        7. ;TRIGER
        1000. ;CLOCKF
        173776 ;ROMADD
        -1000. ;SEC1
        -5000. ;SECS
        -10000. ;SEC10

;MISC SCRATCH PAD
FEEDBK: 24. ;SCHMIDT TRIGGER TIME
RLYPT: 5. ;RELAY PICK TIME
WDPRT: LEVEL6
FIRST.TST:0
TIMRUN: 2000. ;REAL TIME OF 2 SEC. USING MNCLK
SAVR1: 0
SAVR2: 0
    
```

896	001270	000000	TIMOUT:0	
897	001272	000000	TRIGER:0	
898	001274	00000J	CLOCKF:0	
899	001276	000000	DECNUM:0	
900	001300	000000	TTRIG:0	
901	001302	000000	LESS1: 0	
902	001304	000000	XCNT:0	
903	001306	000000	WDBASE: 0	
904	001310	000000	WDIBAS: 0	
905	001312	000000	LOCTRP: 0	
906	001314	000026	TRANST: 22.	; SAVE RESTART ADDRESS HERE
907	001316	176030	SEC1: -1000.	; TRANSISTOR SATURATE TIME
908	001320	166170	SEC5: -5000.	; DEFAULT VALUE
909	001322	154360	SEC10: -10000.	
910	001324	011610	MNCLKT: 5000.	; DEFAULT VALUE-MNCLK TIME=5 SEC.
911	001326	000000	BUFF2: 0	
912	001330	000000	BUFF1: 0	
913	001332	173776	ROMADD: 173776	; DEFAULT RESTART ADDRESS
914				
915				
916				

```

917 ;*****
918 ;TEST 1 W D LINE CLOCK BIT TESTS
919 ;*****
920 001334 104400 TST1: SCOPE
921 001236 012767 000010 011310 MOV #10,ICOUNT ;ITERATION COUNT
922 001344 012767 000001 011160 MOV #1,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
923 001352 012767 001360 011300 MOV #SCP1,RETURN ;SCOPE LOOP RETURN ADRS
924 001360 SCP1:
925
926 :TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
927 001360 032777 000100 177616 LKTEST: BIT #CLKEN,@CLKCSR
928 001366 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
929 001370 104000 ERROR ;ERROR, CLOCK INTERRUPTS ENABLE NOT CLEARED BY START
930 :TEST THAT START SETS CLOCK TICK
931 001372 105777 177606 LK1: TSTB @CLKCSR
932 001376 100401 BMI .+4 ;BRANCH IF NO ERROR CONDITION
933 001400 104000 ERROR ;ERROR, CLOCK TICK NOT SET BY START
934 :TEST THAT CLOCK TICK WILL SET AFTER SUFFICIENT PERIOD OF TIME
935 001402 005077 177576 LK2: CLR @CLKCSR
936 001406 005000 CLR R0
937 001410 105777 177570 LK2A: TSTB @CLKCSR ;IS CLOCK TICK SET
938 001414 100403 BMI LK3
939 001416 005200 INC R0 ;NO, INCREMENT COUNT
940 001420 001373 BNE LK2A ;WAIT SUFFICIENT AMOUNT OF TIME FOR CLOCK
941 001422 104000 ERROR ;ERROR,CLOCK TICK FAILED TO SET
942 :TEST THAT INTERRUPT ENABLE BIT MAY BE SET
943 001424 005077 177554 LK3: CLR @CLKCSR
944 001430 105777 177550 TSTB @CLKCSR
945 001434 100375 BPL .-4 ;SYNC CLOCK
946 001436 012767 000340 176332 MOV #LEVEL7,PS ;SET PRIORITY 7
947 001444 012777 000100 177532 MOV #100,@CLKCSR ;CLEAR CLOCK TICK AND SET INTERRUPT ENABLE
948 001452 032777 000100 177524 BIT #CLKEN,@CLKCSR ;IS INTERRUPT ENABLE SET?
949 001460 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
950 001462 104000 ERROR ;INTERRUPT ENABLE NOT SET
951 :TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
952 001464 005077 177514 LK4: CLR @CLKCSR
953 001470 105777 177510 TSTB @CLKCSR
954 001474 100375 BPL .-4 ;SYNC CLOCK
955 001476 012777 000100 177500 MOV #100,@CLKCSR ;CLEAR CLOCKFLAG AND SET INTERRUPT ENABLE
956 001504 005077 177474 CLR @CLKCSR ;CLEAR INTERRUPT ENABLE
957 001510 032777 000100 177466 BIT #CLKEN,@CLKCSR
958 001516 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
959 001520 104000 ERROR ;INTERRUPT ENABLE NOT CLEARED
960 001522 105777 177456 TSTB @CLKCSR ;WAIT FOR CLOCK TICK
961 001526 100375 BPL .-4
962
963
    
```

```

964                                     ;*****
965                                     ;TEST 2      INTERRUPT ENABLE TEST
966                                     ;*****
967 001530 104400                       TST2: SCOPE
968 001532 012767 000010 011114         MOV #10,ICOUNT ;ITERATION COUNT
969 001540 012767 000002 010764         MOV #2,ERTSTN ;SAVE TEST # FOR ERROR REPORT
970 001546 012767 001554 011104         MOV #SCP2,RETURN ;SCOPE LOOP RETURN ADRS
971 001554                               SCP2:
972
973                                     ;TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
974 001554 016777 177476 177414 LK5: MOV WDPRT,@CLKIS
975 001562 016767 177514 176206         MOV LESS1,PS ;SET PROCESSOR PRIORITY LESS THAN LINE CLOCK
976 001570 012777 001630 177376         MOV #LK5B,@CLKIV ;SET UP VECTOR RETURN POINTER
977 001576 005077 177402
978 001602 105777 177376
979 001606 100375
980 001610 012777 000100 177366         MOV #100,@CLKCSR ;SYNC CLOCK
981 001616 005000                               CLR RD ;ENABLE INTERRUPT AND CLEAR CLOCK TICK
982 001620 005200 LK5A: INC RD
983 001622 001376 BNE LK5A ;WAIT FOR INTERRUPT
984 001624 104000 ERROR ;INTERRUPT FAILED TO OCCUR
985 001626 000433 BR LK6 ;
986 001630 016777 177342 177336 LK5B: MOV CLKIS,@CLKIV ;RESTORE TRAPCATCHER
987 001636 012777 000000 177332         MOV #0,@CLKIS ;
988 001644 105777 177334 TSTB @CLKCSR
989 001650 100401 BMI .+4 ;BRANCH IF NO ERROR CONDITION
990 001652 104000 ERROR ;INTERRUPT NOT CAUSED BY CLOCK
991 001654 012716 001670 MOV #LK5.1,(SP) ;LOAD RETURN ADDRESS
992 001660 012767 000340 176110         MOV #LEVEL7,PS ;RESTORE PROC STATUS
993 001666 000002 RTI
994 001670 000240 LK5.1: NOP
995
996

```



# L02

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 25  
 DBGAAA.P11 T3 CHECK CORRECT COND. CODES AND PROGRAM COUNTER ON STACK

```

997                                     ;*****
998                                     ;TEST 3 CHECK CORRECT COND. CODES AND PROGRAM COUNTER ON STACK
999                                     ;*****
1000 001672 104400 TST3: SCOPE
1001 001674 012767 000010 010752 MOV #10,ICOUNT ; ITERATION COUNT
1002 001702 012767 000003 010622 MOV #3,ERTSTN ; SAVE TEST # FOR ERROR REPORT
1003 001710 012767 001716 010742 MOV #SCP3,RETURN ; SCOPE LOOP RETURN ADRS
1004 001716 SCP3:
1005
1006                                     ;TEST THAT CLOCK INTERRUPT PUSHES COND. CODES AND PROGRAM COUNTER ONTO STACK
1007 001716 016777 177334 177252 LK6: MOV WDPRT,CLKIS
1008 001724 005077 177254 CLR CLKCSR
1009 001730 105777 177250 TSTB CLKCSR
1010 001734 100375 BPL -4 ; SYNC CLOCK
1011 001736 012777 001764 177230 MOV #LK6A,CLKIV ; SET UP VECTOR RETURN
1012 001744 016767 177332 176024 MOV LESS1,PS ; SET PROC. PRIORITY
1013 001752 012777 000100 177224 MOV #100,CLKCSR ; SET ENABLE BIT
1014 001760 000277 SCC ; SET ALL COND. CODES
1015 001762 000001 WAIT ; WAIT FOR INTERRUPT
1016 001764 016777 177206 177202 LK6A: MOV CLKIS,CLKIV ; RESTORE CLKIV
1017 001772 012777 000000 177176 MOV #0,CLKIS
1018 002000 012767 000340 175770 MOV #LEVEL7,PS ; RESTORE PROC. STATUS
1019 002006 011637 001330 MOV (SP),BUFF1 ; PC
1020 002012 016637 000002 001326 MOV 2(SP),BUFF2 ; PS
1021 002020 022737 001764 001330 CMP #LK6A,BUFF1
1022 002026 001401 BEQ +4 ; BRANCH IF NO ERROR CONDITION
1023 002030 104000 ERROR ; WRONG PC PUSHED ONTO STACK
1024 002032 162737 000017 001326 SUB #17,BUFF2 ; SUBTRACT CONDITION CODES
1025 002040 026737 177236 001326 CMP LESS1,BUFF2
1026 002046 001401 BEQ +4 ; BRANCH IF NO ERROR CONDITION
1027 002050 104000 ERROR ; WRONG PS PUSHED ONTO STACK
1028 002052 012716 002060 MOV #LK6.1,(SP) ; LOAD RETURN ADDRESS
1029 002056 000002 RTI
1030 002060 LK6.1: NOP
1031
1032

```

```

1033
1034
1035
1036 002062 104400
1037 002064 012767 000010 010562
1038 002072 012767 000004 010432
1039 002100 012767 002106 010552
1040 002106
1041
1042
1043 002106 005077 177072
1044 002112 105777 177066
1045 002116 100375
1046 002120 005077 177060
1047 002124 000005
1048 002126 105777 177052
1049 002132 100401
1050 002134 104000
1051
1052

```

```

;*****
;TEST 4 RESET TEST
;*****
TST4: SCOPE
MOV #10,ICOUNT ; ITERATION COUNT
MOV #4,ERTSTN ; SAVE TEST # FOR ERROR REPORT
MOV #SCP4,RETURN ; SCOPE LOOP RETURN ADRS
SCP4:
;TEST THAT RESET SETS CLOCK FLAG
LK7: CLR @CLKCSR ; CLEAR CLOCK FLAG
TSTB @CLKCSR
BPL -4 ; WAIT FOR CLOCK TICK
CLR @CLKCSR ; SHOULD CLEAR CLOCK TICK
RESET
TSTB @CLKCSR ; BRANCH IF NO ERROR CONDITION
BMI .+4 ; RESET DID NOT CLEAR CLOCK FLAG
ERROR

```

```

1053 ;*****
1054 ;TEST 5 LINE CLOCK REPEATIBILITY
1055 ;*****
1056 002136 104400 TST5: SCOPE
1057 002140 012767 000010 010506 MOV #10,ICOUNT ;ITERATION COUNT
1058 002146 012767 000005 010356 MOV #5,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
1059 002154 012767 002162 010476 MOV #SCP5,RETURN ;SCOPE LOOP RETURN ADRS
1060 002162 SCP5:
1061 ;TEST LINE CLOCK REPEATABILITY
1062 ;MAKE SURE THAT OVER TWO EQUAL PERIODS OF TIME
1063 ;THE LINE CLOCK PUTS OUT THE SAME NUMBER OF PULSES
1064 LK8: CLR R0 ;CLEAR 1ST TIME COUNT
1065 002162 005000 CLR R1 ;CLEAR 1ST CLOCK COUNT
1066 002164 005001 MOV #LEVEL7,PS ;SET PRIORITY
1067 002166 012767 000340 175602 CLR @CLKCSR
1068 002174 005077 177004 TSTB @CLKCSR
1069 002200 105777 177000 BPL -4 ;SYNC CLOCK
1070 002204 100375 CLR @CLKCSR
1071 002206 005077 176772 TSTB @CLKCSR
1072 002212 105777 176766 BPL -4 ;SYNC CLOCK 2ND TIME TO MAKE SURE
1073 002216 100375 CLR @CLKCSR
1074 002220 005077 176760 LK8A: TSTB @CLKCSR ;IS CLOCK TICK SET
1075 002224 105777 176754 BPL LK8B ;NO
1076 002230 100003 INC R1 ;+1 TO CLOCK COUNT
1077 002232 005201 CLR @CLKCSR ;CLEAR CLOCK IF SET
1078 002234 005077 176744 LK8B: INC R0 ;+1 TO TIME COUNT
1079 002240 005200 BNE LK8A ;REPEAT UNTIL R0=0
1080 002242 001370 CLR R0 ;CLEAR 2ND TIME COUNT
1081 002244 005000 CLR R2 ;CLEAR 2ND CLOCK COUNT
1082 002246 005002 CLR @CLKCSR
1083 002250 005077 176730 TSTB @CLKCSR
1084 002254 105777 176724 BPL -4
1085 002260 100375 CLR @CLKCSR
1086 002262 005077 176716 TSTB @CLKCSR
1087 002266 105777 176712 BPL -4 ;SYNC CLOCK 2ND TIME TO MAKE SURE
1088 002272 100375 CLR @CLKCSR
1089 002274 005077 176704 LK8C: TSTB @CLKCSR ;IS CLOCK TICK SET
1090 002300 105777 176700 BPL LK8D ;NO
1091 002304 100003 INC R2 ;+1 TO CLOCK COUNT
1092 002306 005202 CLR @CLKCSR ;CLEAR CLOCK IF SET
1093 002310 005077 176670 LK8D: INC R0 ;+1 TO TIME COUNT
1094 002314 005200 BNE LK8C ;REPEAT UNTIL R0=0
1095 002316 001370 MOV R1,SAVR1 ;SAVE R1
1096 002320 010167 176740 MOV R2,SAVR2 ;SAVE R2
1097 002324 010267 176736 CMP R1,R2 ;ARE THEY EQUAL ?
1098 002330 020102 BEQ LK8E ;YES
1099 002332 001411 ADD #1,R1 ;ARE THEY OFF BY 1 COUNT ?
1100 002334 062701 000001 CMP R1,R2 ;YES
1101 002340 020102 BEQ LK8E ;ARE THEY OFF BY 1 COUNT ?
1102 002342 001405 SUB #2,R1 ;YES
1103 002344 162701 000002 CMP R1,R2 ;CLOCK TICK DIFF BY MORE THAN 1 COUNT
1104 002350 020102 BEQ LK8E ;IN EQUAL PERIODS OF TIME
1105 002352 001401 ERROR
1106 002354 104000 LK8E:
1107 1107
1108 002356 000240 LK8E: NOP
    
```

B03

MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 28  
DBQAAA.P11 TS LINE CLOCK REPEATIBILITY

1109  
1110

```

1111 ;*****
1112 ;TEST 6 LINE CLOCK REPEATABILITY WITH INTERRUPT
1113 ;*****
1114 002360 104400 TST6: SCOPE
1115 002362 012767 000010 010264 MOV #10,ICOUNT ;ITERATION COUNT
1116 002370 012767 000006 010134 MOV #6,ERTSTN ;SAVE TEST # FOR ERROR REPORT
1117 002376 012767 002404 010254 MOV #SCP6,RETURN ;SCOPE LOOP RETURN ADRS
1118 002404 SCP6:
1119
1120 ;TEST LINE CLOCK REPEATABILITY WITH INTERRUPT
1121 002404 016767 176672 175364 LK9: MOV LESS1,PS ;SET PROC. PRIORITY
1122 002412 005001 CLR R1 ;CLEAR 1ST CLOCK COUNT
1123 002414 012777 002500 176552 MOV #LK9B,@CLKIV
1124 002422 016777 176630 176546 MOV WDPRT,@CLKIS
1125 002430 005077 176550 CLR @CLKCSR
1126 002434 105777 176544 TSTB @CLKCSR
1127 002440 100375 BPL -4 ;SYNC CLOCK
1128 002442 005077 176536 CLR @CLKCSR
1129 002446 105777 176532 TSTB @CLKCSR
1130 002452 100375 BPL -4 ;SYNC CLOCK 2ND TIME TO MAKE SURE
1131 002454 012777 000100 176522 MOV #100,@CLKCSR ;ENABLE INTERRUPT
1132 ;WAIT FOR CLOCK INTERRUPTS
1133 002462 004567 005632 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1134 002466 001262 TIMRUN ;FUNCTION TO OCCUR
1135 002470 042777 000100 176506 BIC #CLKEN,@CLKCSR ;CLEAR INTERRUPT ENABLE
1136 002476 000405 BR LK9C
1137 002500 005201 LK9B: INC R1
1138 002502 012777 000100 176474 MOV #100,@CLKCSR ;CLEAR TICK & SET ENABLE
1139 002510 000002 RTI
1140 002512 016767 176564 175256 LK9C: MOV LESS1,PS
1141 002520 005002 CLR R2 ;CLEAR 2ND CLOCK COUNT
1142 002522 012777 002606 176444 MOV #LK9E,@CLKIV
1143 002530 016777 176522 176440 MOV WDPRT,@CLKIS
1144 002536 005077 176442 CLR @CLKCSR
1145 002542 105777 176436 TSTB @CLKCSR
1146 002546 100375 BPL -4 ;SYNC CLOCK
1147 002550 005077 176430 CLR @CLKCSR
1148 002554 105777 176424 TSTB @CLKCSR
1149 002560 100375 BPL -4 ;SYNC CLOCK SECOND TIME TO MAKE SURE
1150 002562 012777 000100 176414 MOV #100,@CLKCSR ;ENABLE INTERRUPT
1151 ;WAIT FOR CLOCK INTERRUPTS
1152 002570 004567 005524 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1153 002574 001262 TIMRUN ;FUNCTION TO OCCUR
1154 002576 042777 000100 176400 BIC #CLKEN,@CLKCSR ;CLEAR INTERRUPT ENABLE
1155 002604 000405 BR LK9F
1156 002606 005202 LK9E: INC R2
1157 002610 012777 000100 176366 MOV #100,@CLKCSR ;CLEAR TICK & SET ENABLE
1158 002616 000002 RTI
1159 002620 005077 176360 LK9F: CLR @CLKCSR
1160 002624 012767 000340 175144 MOV #LEVEL7,PS ;RESTORE PROC. STATUS
1161 002632 016777 176340 176334 MOV CLKIS,@CLKIV ;RESTORE CLKIV
1162 002640 012777 000000 176330 MOV #0,@CLKIS
1163 002646 010167 176412 MOV R1,SAVR1 ;SAVE
1164 002652 010267 176410 MOV R2,SAVR2 ;SAVE
1165 002656 020127 000000 CMP R1,#0
1166 002662 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION

```

1167	002664	104000		ERROR			
1168	002666	020127	000001	CMP	R1, #1		;PROC. IS NOT ALLOWING INTERRUPTS
1169	002672	001001		BNE	.+4		;BRANCH IF NO ERROR CONDITION
1170	002674	104000		ERROR			;PROC. IS NOT ALLOWING INTERRUPTS
1171	002676	020102		CMP	R1, R2		;ARE THEY EQUAL ?
1172	002700	001411		BEQ	LK9G		
1173	002702	062701	000001	ADD	#1, R1		;ARE THEY OFF BY 1 COUNT ?
1174	002706	020102		CMP	R1, R2		
1175	002710	001405		BEQ	LK9G		
1176	002712	162701	000002	SUB	#2, R1		;ARE THEY OFF BY 1 COUNT ?
1177	002716	020102		CMP	R1, R2		
1178	002720	001401		BEQ	LK9G		
1179	002722	104000		ERROR			;CLOCK TICK DIFF. BY MORE THAN 1 COUNT
1180							;IN EQUAL PERIODS OF TIME
1181	002724	000240		LK9G:	NOP		
1182							
1183							

# E03

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 31  
 DBQAAA.P11 T7 LINE CLOCK ACCURACY TEST

```

1184 :*****
1185 :TEST 7 LINE CLOCK ACCURACY TEST
1186 :*****
1187 002726 104400 ST7: SCOPE
1188 002730 012767 000000 007716 MOV #0,ICOUNT ; ITERATION COUNT
1189 002736 012767 000007 007566 MOV #7,ERTSTN ; SAVE TEST # FOR ERROR REPORT
1190 002744 012767 002752 007706 MOV #SCP7,RETURN ; SCOPE LOOP RETURN ADRS
1191 002752 SCP7:
1192
1193 ;RING BELL AFTER EACH OF FIRST (5) 1 SECOND INTERUALS
1194 ;RING BELL AFTER 10 SECOND INTERVAL
1195 ;(INTERRUPT MODE)
1196 002752 016700 176340 LK10: MOV SEC1,RO
1197 002756 004767 005762 JSR PC,TIME ;BELL AFTER 1 SECOND
1198 002762 016700 176330 MOV SEC1,RO
1199 002766 004767 005752 JSR PC,TIME ;BELL AFTER 2 SEC.
1200 002772 016700 176320 MOV SEC1,RO
1201 002776 004767 005742 JSR PC,TIME ;BELL AFTER 3 SEC.
1202 003002 016700 176310 MOV SEC1,RO
1203 003006 004767 005732 JSR PC,TIME ;BELL AFTER 4 SEC.
1204 003012 016700 176300 MOV SEC1,RO
1205 003016 004767 005722 JSR PC,TIME ;BELL AFTER 5 SEC.
1206 003022 016700 176272 MOV SEC5,RO
1207 003026 004767 005712 JSR PC,TIME ;BELL AFTER 10 SEC.
1208 ;RING BELL AFTER 10 SECONDS USING CLOCK TICK (NON INTERRUPT)
1209 ;5 TIMES
1210 003032 012703 177773 MOV #5,R3
1211 003036 016700 176260 LK10B: MOV SEC10,RO
1212 003042 005077 176136 LK10A: CLR @CLKCSR ;CLEAR CLOCK TICK
1213 003046 105777 176132 TSTB @CLKCSR ;WAIT FOR CLOCK TICK
1214 003052 100375 BPL -4
1215 003054 005200 INC RO ;FINISHED 10 SECONDS?
1216 003056 001371 BNE LK10A
1217 003060 000004 012206 TYPE BELL ;RING BELL EVERY 10 SECONDS
1218 003064 005203 INC R3
1219 003066 001363 BNE LK10B ;5 BELLS
1220
1221
  
```

# F03

MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 32  
 DBQAAA.P11 T10 MNCLK TEST

```

1222 ;*****
1223 ;TEST 10 MNCLK TEST
1224 ;*****
1225 003070 104400 TST10: SCOPE
1226 003072 012767 000010 007554 MOV #10,ICOUNT ;ITERATION COUNT
1227 003100 012767 000010 007424 MOV #10,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
1228 003106 012767 003114 007544 MOV #SCP10,RETURN ;SCOPE LOOP RETURN ADRS
1229 003114 SCP10:
1230
1231 ;MAINTENANCE CLOCK BIT TEST--MNCLK
1232 003114 005000 LK11: CLR R0 ;CLEAR 0 STATE COUNTER
1233 003116 005001 CLR R1 ;CLEAR 1 STATE COUNTER
1234 003120 012702 000001 MOV #1,R2 ;TIME COUNTER
1235 003124 032777 000010 176060 TMNCLK: BIT #MNCLK,AWDDAT ;TEST MNCLK BIT
1236 003132 001002 BNE SET
1237 003134 005200 CLEAR: INC R0 ;STORE 0 STATE
1238 003136 000401 BR .+4
1239 003140 005201 SET: INC R1 ;STORE 1 STATE
1240 003142 005202 INC R2
1241 003144 001367 BNE TMNCLK ;WAIT SUFFICIENT TIME TO STORE BOTH STATES
1242 003146 005700 TST R0
1243 003150 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1244 003152 104000 ERROR ;MNCLK IS NOT WORKING-CLEARED
1245 003154 005701 TST R1
1246 003156 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1247 003160 104000 ERROR ;MNCLK IS NOT WORKING-SET
1248
1249
  
```



```

1250 ;*****
1251 ;TEST 11 MNCLK ACCURACY TEST
1252 ;*****
1253 003162 104400 TST11: SCOPE
1254 003164 012767 000000 007462 MOV #0,ICOUNT ;ITERATION COUNT
1255 003172 012767 000011 007332 MOV #11,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
1256 003200 012767 003206 007452 MOV #SCP11,RETURN ;SCOPE LOOP RETURN ADRS
1257 003206 SCP11:
1258
1259 ;RING BELL AFTER 5 SEC INTERVAL--5 TIMES
1260 003206 012703 177773 LK12: MOV #-5,R3
1261 003212 005077 175766 CLR @CLKCSR
1262 003216 105777 175762 TSTB @CLKCSR
1263 003222 100375 BPL .-4 ;SYNC CLOCK
1264 003224 LK12A:
1265 003224 004567 005070 JSR R5,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1266 003230 001324 MNCLKT ;FUNCTION TO OCCJR
1267 003232 000004 012206 TYPE BELL ;BELL
1268 003236 005203 INC R3
1269 003240 001371 BNE LK12A ;RING BELL 5 TIMES
1270
1271

```

# H03

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 34  
DBGAAA.P11 T12 WD1,WD2 READ/WRITE TEST

```
1272 ;*****
1273 ;TEST 12 WD1,WD2 READ/WRITE TEST
1274 ;*****
1275 003242 104400 TST12: SCOPE
1276 003244 012767 000010 007402 MOV #10,ICOUNT ;ITERATION COUNT
1277 003252 012767 000012 007252 MOV #12,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
1278 003260 012767 003266 007372 MOV #SCP12,RETURN ;SCOPE LOOP RETURN ADRS
1279 003266 SCP12:
1280
1281
1282 ;THE FOLLOWING JUMPERS MUST BE INSERTED
1283 ;IF JUMPER PLUG IS USED --OPEN SWITCH--
1284 ;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
1285 ;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
1286 003266 005077 175716 CLR @WDCSR ;CLEAR WDCSR
1287 003272 117727 175712 MOVB @WDCSR,(PC)+
1288 003276 000000 STORE2: 0 ;PASS CONTENTS OF WDCSR INTO HERE
1289 003300 042737 000020 003276 BIC #20,@STORE2 ;CLEAR REMSEN BIT
1290 003306 122737 000000 003276 CMPB #0,@STORE2
1291 003314 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1292 003316 104000 ERROR ;WDCSR NOT CLEARED
1293 003320 052777 000002 175664 BIS #WD1,@WDDAT ;SET WD1
1294 003326 032777 000002 175656 BIT #WD1,@WDDAT ;VERIFY SET
1295 003334 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1296 003336 104000 ERROR ;WD1 NOT SET
1297 003340 032777 000004 175644 BIT #WD2,@WDDAT ;WD1 SHOULD NOT BE CLEARED, WD2 MUST BE SET
1298 003346 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1299 003350 104000 ERROR ;WD2 NOT SET
1300 003352 042777 000004 175632 BIC #WD2,@WDDAT ;CLEAR WD2 ,WD2 CLEARS WD1 AND THEN WD1 SETS WD2
1301 003360 032777 000002 175624 BIT #WD1,@WDDAT
1302 003366 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1303 003370 104000 ERROR ;WD1 NOT CLEARED
1304 003372 032777 000004 175612 BIT #WD2,@WDDAT
1305 003400 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1306 003402 104000 ERROR ;WD2 NOT SET
1307
1308
1309
```

```

1310 ;*****
1311 ;TEST 13 TOO SOON
1312 ;*****
1313 003404 104400 TST13: SCOPE
1314 003406 012767 000010 007240 MOV #10,ICOUNT ;ITERATION COUNT
1315 003414 012767 000013 007110 MOV #13,ERTSTM ;SAVE TEST # FOR ERROR REPCRT
1316 003422 012767 003430 007230 MOV #SCP13,RETURN ;SCOPE LOOP RETURN ADRS
1317 003430 SCP13:
1318
1319
1320 ;THE FOLLOWING JUMPERS MUST BE INSERTED
1321 ;IF JUMPER PLUG IS USED --OPEN SWITCH--
1322 ;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
1323 ;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
1324 003430 005077 175554 CLR #WDCSR ;CLEAR TOOSN ETC.
1325 003434 052777 000002 175550 BIS #WD1,#WDDAT ;SET WD1,FIRST UPDATE SEQ.
1326 003442 032777 000002 175542 BIT #WD1,#WDDAT ;VERIFY SET
1327 003450 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1328 003452 104000 ERROR ;WD1 NOT SET
1329 003454 032777 000004 175530 BIT #WD2,#WDDAT ;WD2 SHOULD NOT BE CLEARED
1330 003462 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1331 003464 104000 ERROR ;WD2 NOT SET
1332 003466 042777 000004 175516 BIC #WD2,#WDDAT ;CLEAR WD2
1333 003474 032777 000002 175510 BIT #WD1,#WDDAT
1334 003502 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1335 003504 104000 ERROR ;WD1 NOT CLEARED
1336 003506 032777 000004 175476 BIT #WD2,#WDDAT
1337 003514 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1338 003516 104000 ERROR ;WD2 NOT SET
1339
1340 003520 052777 000002 175464 BIS #WD1,#WDDAT ;SECOND UPDATE SEQ.
1341 003526 032777 000002 175456 BIT #WD1,#WDDAT ;VERIFY SET
1342 003534 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1343 003536 104000 ERROR ;WD1 NOT SET
1344 003540 032777 000004 175444 BIT #WD2,#WDDAT ;WD2 SHOULD NOT BE CLEARED
1345 003546 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1346 003550 104000 ERROR ;WD2 NOT SET
1347 003552 042777 000004 175432 BIC #WD2,#WDDAT ;CLEAR WD2
1348 003560 032777 000002 175424 BIT #WD1,#WDDAT
1349 003566 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1350 003570 104000 ERROR ;WD1 NOT CLEARED
1351 003572 032777 000004 175412 BIT #WD2,#WDDAT
1352 003600 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1353 003602 104000 ERROR ;WD2 NOT SET
1354 003604 032777 000040 175376 BIT #TOOSN,#WDCSR
1355 003612 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1356 003614 104000 ERROR ;TOO SOON DID NOT SET
1357 003616 042777 000040 175364 BIC #TOOSN,#WDCSR
1358 003624 032777 000040 175356 BIT #TOOSN,#WDCSR
1359 003632 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1360 003634 104000 ERROR ;TOO SOON IS NOT CLEARED
1361
1362

```

```

1363 :*****
1364 :TEST 14 WDCSR READ/WRITE BIT TEST
1365 :*****
1366 003636 104400
1367 003640 012767 000010 007006
1368 003646 012767 000014 006656
1369 003654 012767 003662 006776
1370 003662
1371
1372
1373
1374
1375
1376
1377 003662 052777 000100 175320
1378 003670 032777 000100 175312
1379 003676 001001
1380 003700 104000
1381 003702 042777 000100 175300
1382 003710 032777 000100 175272
1383 003716 001401
1384 003720 104000
1385
1386 003722 052777 000010 175260
1387 003730 032777 000010 175252
1388 003736 001001
1389 003740 104000
1390 003742 042777 000010 175240
1391 003750 032777 000010 175232
1392 003756 001401
1393 003760 104000
1394
1395 003762 052777 000004 175220
1396 003770 032777 000004 175212
1397 003776 001001
1398 004000 104000
1399 004002 042777 000004 175200
1400 004010 032777 000004 175172
1401 004016 001401
1402 004020 104000
1403
1404

;*****
;TEST 14 WDCSR READ/WRITE BIT TEST
;*****
;ST14: SCOPE
;MOV #10,ICOUNT ; ITERATION COUNT
;MOV #14,ERTSTN ; SAVE TEST # FOR ERROR REPORT
;MOV #SCP14,RETURN ; SCOPE LOOP RETURN ADRS
SCP14:

;THE FOLLOWING JUMPERS MUST BE INSERTED
;IF JUMPER PLUG IS USED --OPEN SWITCH--
;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
;BIS #WDIEN,WDCSR
;BIT #WDIEN,WDCSR
;BNE .+4 ; BRANCH IF NO ERROR CONDITION
;WD INTERRUPT ENABLE CANNOT BE SET
;BIC #WDIEN,WDCSR
;BIT #WDIEN,WDCSR
;BEQ .+4 ; BRANCH IF NO ERROR CONDITION
;WD INTERRUPT ENABLE BE CLEARED
;BIS #EXTREQ,WDCSR
;BIT #EXTREQ,WDCSR
;BNE .+4 ; BRANCH IF NO ERROR CONDITION
;EXTREQ CANNOT BE SET
;BIC #EXTREQ,WDCSR
;BIT #EXTREQ,WDCSR
;BEQ .+4 ; BRANCH IF NO ERROR CONDITION
;EXTREQ CANNOT BE CLEARED
;WD ENABLE OPTIONAL JUMPER MUST BE INSTALLED
;BIS #WDENAB,WDCSR
;BIT #WDENAB,WDCSR
;BNE .+4 ; BRANCH IF NO ERROR CONDITION
;WD ENABLE CANNOT BE SET,CHECK JUMPER
;BIC #WDENAB,WDCSR
;BIT #WDENAB,WDCSR
;BEQ .+4 ; BRANCH IF NO ERROR CONDITION
;WD ENABLE CANNOT BE CLEARED
  
```

```

1405 ;*****
1406 ;TEST 15 INTERRUPT ENABLE TEST
1407 ;*****
1408 004022 104400 TST15: SCOPE
1409 004024 012767 000010 006622 MOV #10,ICOUNT ;ITERATION COUNT
1410 004032 012767 000015 006472 MOV #15,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
1411 004040 012767 004046 006612 MOV #SCP15,RETURN ;SCOPE LOOP RETURN ADRS
1412 004046 SCP15:
1413
1414
1415 ;THE FOLLOWING JUMPERS MUST BE INSERTED
1416 ;IF JUMPER PLUG IS USED --OPEN SWITCH--
1417 ;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
1418 ;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
1419 004046 005077 175136 CLR #WDCSR
1420 004052 100001 BPL .+4 ;BRANCH IF NO ERROR CONDITION
1421 004054 104000 ERROR ;TOO SOON IS NOT CLEARED
1422 004056 016777 175174 175116 MOV WDPRT,#WDIS
1423 004064 012777 004162 175106 MOV #INTTST,#WDIV
1424 004072 016767 175204 173676 MOV LESS1,#PS
1425 004100 032777 000002 175102 BIT #WDRLY,#WDCSR
1426 004106 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1427 004110 104000 ERROR ;WD RELAY SHOULD NOT BE SET
1428 004112 052777 000100 175070 BIS #WDIEN,#WDCSR
1429 004120 052777 000002 175064 BIS #WD1,#WDDAT ;UPDATE SEQ.
1430 004126 042777 000004 175056 BIC #WD2,#WDDAT
1431 004134 052777 000002 175050 BIS #WD1,#WDDAT ;THIS SECOND UPDATE SEQUENCE SHOULD
1432 ;SET TOO SOON AND SET FLAG
1433 004142 042777 000004 175042 BIC #WD2,#WDDAT
1434
1435 004150 005000 CLR RO
1436 004152 005200 DELAY: INC RO ;WAIT UNTIL INTERRUPT
1437 004154 001376 BNE DELAY
1438 004156 104000 ERROR ;INTERRUPT DID NOT OCCUR
1439 004160 000435 BR CONTI
1440 004162 016777 175014 175010 INTTST: MOV WDIS,#WDIV ;RESTORE WDIV
1441 004170 012777 000000 175004 MOV #0,#WDIS
1442 004176 105777 175006 TSTB #WDCSR
1443 004202 100401 BMI .+4 ;TEST FLAG
1444 004204 104000 ERROR ;BRANCH IF NO ERROR CONDITION
1445 004206 032777 000040 174774 BIT #TOOSN,#WDCSR ;ERROR ,INTERRUPT NOT CAUSED BY FLAG
1446 004214 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1447 004216 104000 ERROR ;TOO SOON DID NOT GET SET
1448 004220 042777 000040 174762 BIC #TOOSN,#WDCSR
1449 004226 032777 000040 174754 BIT #TOOSN,#WDCSR
1450 004234 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1451 004236 104000 ERROR ;TOO SOON DID NOT CLEAR
1452 004240 012716 004254 MOV #CONTI,(SP) ;LOAD RETURN ADDRESS
1453 004244 012767 000340 173524 MOV #LEVEL7,#PS ;RESTORE PROC STATUS
1454 004252 000002 RTI
1455
1456 004254 042777 000004 174726 CONTI: BIC #WDENAB,#WDCSR ;CLEAR WDENAB--THIS DROPS WDRLY
1457 004262 005027 CLR (PC)+ ;WAITING SO THAT WD RELAY
1458 ;CLEARS BEFORE PROCEEDING
1459 004264 000000 15: 0
1460 004266 005367 177772 25: DEC 15
  
```

L03

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 38  
DBQAAA.P11 T15 INTERRUPT ENABLE TEST

1461 004272 001375  
1462  
1463

BNE 25

```

1464 ;*****
1465 ;TEST 16 REVERSE UPDATE SEQUENCE TEST
1466 ;*****
1467 004274 104400 TST16: SCOPE
1468 004276 012767 000010 006350 MOV #10,ICOUNT ; ITERATION COUNT
1469 004304 012767 000016 006220 MOV #16,ERTSTN ; SAVE TEST # FOR ERROR REPCRT
1470 004312 012767 004320 006340 MOV #SCP16,RETURN ; SCOPE LOOP RETURN ADRS
1471 004320 SCP16:
1472
1473 ; THE FOLLOWING JUMPERS MUST BE INSERTED
1474 ; IF JUMPER PLUG IS USED --OPEN SWITCH--
1475 ; PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
1476 ; PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
1477 004320 005077 174664 CLR @WDCSR
1478 004324 005077 174662 CLR @WDDAT ; WD2 WILL ALWAYS BE "SEEN" SET
1479 004330 032777 000002 174654 BIT #WD1,@WDDAT
1480 004336 001401 BEQ .+4 ; BRANCH IF NO ERROR CONDITION
1481 004340 104000 ERROR ; WD1 IS NOT CLEARED
1482 004342 032777 000004 174642 BIT #WD2,@WDDAT
1483 004350 001001 BNE .+4 ; BRANCH IF NO ERROR CONDITION
1484 004352 104000 ERROR ; WD2 IS NOT SET
1485 004354 012777 000002 174630 MOV #2,@WDDAT ; SET WD1 & TRY TO CLEAR WD2 SIMULTANEOUSLY
1486 004362 032777 000002 174622 BIT #WD1,@WDDAT
1487 004370 001001 BNE .+4 ; BRANCH IF NO ERROR CONDITION
1488 004372 104000 ERROR ; WD1 IS NOT SET
1489 004374 032777 000004 174610 BIT #WD2,@WDDAT
1490 004402 001001 BNE .+4 ; BRANCH IF NO ERROR CONDITION
1491 004404 104000 ERROR ; WD2 IS NOT SET
1492 004406 012777 000002 174576 MOV #2,@WDDAT ; SET WD1 & TRY TO CLEAR WD2 SIMULTANEOUSLY
1493 004414 032777 000002 174570 BIT #WD1,@WDDAT
1494 004422 001401 BEQ .+4 ; BRANCH IF NO ERROR CONDITION
1495 004424 104000 ERROR ; WD1 IS NOT CLEARED
1496 004426 032777 000004 174556 BIT #WD2,@WDDAT
1497 004434 001001 BNE .+4 ; BRANCH IF NO ERROR CONDITION
1498 004436 104000 ERROR ; WD2 IS NOT SET
1499 ;WAIT UNTIL LOOP CLEARS
1500 004440 005027 CLR (PC)+
1501 004442 000000 2$: 0
1502 004444 005367 177772 3$: DEC 2$
1503 004450 001375 BNE 3$
1504
1505

```

```

1506 ;*****
1507 ;TEST 17 RELAY PICK TEST
1508 ;*****
1509 004452 104400 TST17: SCOPE
1510 004454 012767 000010 006172 MOV #10,ICOUNT ; ITERATION COUNT
1511 004462 012767 000017 006042 MOV #17,ERTSTN ; SAVE TEST # FOR ERROR REPORT
1512 004470 012767 004476 006162 MOV #SCP17,RETURN ; SCOPE LOOP RETURN ADRS
1513 004476 SCP17:
1514
1515 ; THE FOLLOWING JUMPERS MUST BE INSERTED
1516 ; IF JUMPER PLUG IS USED --OPEN SWITCH--
1517 ; PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
1518 ; PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
1519 004476 005077 174506 CLR @WDCSR
1520 004502 032777 000002 174500 BIT #WDRLY,@WDCSR
1521 004510 001401 BEQ .+4 ; BRANCH IF NO ERROR CONDITION
1522 004512 104000 ERROR ; WD RELAY SHOULD NOT BE PICKED
1523 004514 052777 000004 174466 BIS #WDENAB,@WDCSR ; ENABLE RELAY
1524 004522 052777 000002 174462 BIS #WD1,@WDDAT ; UPDATE SEQ.
1525 004530 042777 000004 174454 BIC #WD2,@WDDAT
1526 004536 004567 003556 JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
1527 004542 001254 RLYPT ; FUNCTION TO OCCUR
1528 004544 032777 000002 174436 BIT #WDRLY,@WDCSR
1529 004552 001001 BNE .+4 ; BRANCH IF NO ERROR CONDITION
1530 004554 104000 ERROR ; WD RELAY DID NOT PICK
1531 004556 042777 000004 174424 BIC #WDENAB,@WDCSR ; CLEAR ENABLE
1532 004564 004567 003530 JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
1533 004570 001254 RLYPT ; FUNCTION TO OCCUR
1534 004572 032777 000002 174410 BIT #WDRLY,@WDCSR
1535 004600 001401 BEQ .+4 ; BRANCH IF NO ERROR CONDITION
1536 004602 104000 ERROR ; CLEARING WDENAB DID NOT CLEAR ENABLE RELAY
1537 ; WAIT UNTIL TRIGGER O.S. FALLS
1538 004604 004567 003510 JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
1539 004610 001272 TRIGER ; FUNCTION TO OCCUR
1540 ; WAIT AGAIN TO MAKE SURE
1541 004612 004567 003502 JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
1542 004616 001272 TRIGER ; FUNCTION TO OCCUR
1543 004620 052777 000004 174362 BIS #WDENAB,@WDCSR ; ENABLE RELAY
1544 004626 052777 000002 174356 BIS #WD1,@WDDAT ; UP DATE SEQ.
1545 004634 042777 000004 174350 BIC #WD2,@WDDAT
1546 004642 004567 003452 JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
1547 004646 001270 TIMOUT ; FUNCTION TO OCCUR
1548 004650 032777 000002 174332 BIT #WDRLY,@WDCSR
1549 004656 001401 BEQ .+4 ; BRANCH IF NO ERROR CONDITION
1550 004660 104000 ERROR ; TIMEOUT VALUE TOO LOW
1551 ; WAIT UNTIL LOOP CLEARS
1552 004662 005027 CLR (PC)+
1553 004664 000000 2$: 0
1554 004666 005367 177772 3$: DEC 2$
1555 004672 001375 BNE 3$
1556
1557
1558
1559

```



```

1560
1561
1562
1563 004674 104400
1564 004676 012767 000010 005750
1565 004704 012767 000020 005620
1566 004712 012767 004720 005740
1567 004720
1568
1569
1570
1571
1572
1573 004720 005077 174264
1574 004724 032777 000040 174256
1575 004732 001401
1576 004734 104000
1577 004736 052777 000002 174246
1578 004744 042777 000004 174240
1579 004752 004567 003342
1580 004756 001272
1581 004760 052777 000002 174224
1582 004766 042777 000004 174216
1583 004774 032777 000040 174206
1584 005002 001401
1585 005004 104000
1586
1587 005006 004567 003306
1588 005012 001270
1589
1590

```

```

:*****
:TEST 20 TRIGER VALUE TEST--TOO SOON
:*****
†ST20: SCOPE
MOV #10,ICOUNT ;ITERATION COUNT
MOV #20,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
MOV #SCP20,RETURN ;SCOPE LOOP RETURN ADRS
SCP20:
;THE FOLLOWING JUMPERS MUST BE INSERTED
;IF JUMPER PLUG IS USED --OPEN SWITCH--
;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
CLR @WDCSR
BIT @TOOSN,@WDCSR
BEQ .+4 ;BRANCH IF NO ERROR CONDITION
ERROR ;TOOSN SHOULD NOT BE SET
;UP DATE SEQ.
BIS #WD1,@WDDAT
BIC #WD2,@WDDAT
JSR R5,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
;FUNCTION TO OCCUR
TRIGER
BIS #WD1,@WDDAT ;SECOND UP DATE SEQ.
BIC #WD2,@WDDAT
BIT @TOOSN,@WDCSR
BEQ .+4 ;BRANCH IF NO ERROR CONDITION
ERROR ;TRIGER VALUE TOO LOW
;WAIT TO AVOID TOOSN
JSR R5,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
;FUNCTION TO OCCUR
TIMOUT

```

C04

1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646

005014 104400  
 005016 012767 000010 005630  
 005024 012767 000021 005500  
 005032 012767 005040 005620  
 005040  
  
 005040 005077 174144  
 005044 032777 000002 174136  
 005052 001401  
 005054 104000  
 005056 127727 174126 000000  
 005064 001401  
 005066 104000  
 005070 052777 000004 174112  
 005076 032777 000004 174104  
 005104 001001  
 005106 104000  
 005110 005077 174076  
 005114 017727 174072  
 005120 000000  
 005122 042737 000370 005120  
 005130 126727 177764 000004  
 005136 001401  
 005140 104000  
 005142 052777 000002 174042  
 005150 032777 000002 174034  
 005156 001001  
 005160 104000  
 005162 042777 000004 174022  
 005170 032777 000002 174014  
 005176 001401  
 005200 104000  
 005202 004567 003112  
 005206 001252  
 005210 004567 003104  
 005214 001272  
 005216 017727 173766  
 005222 000000

```

:*****
:TEST 21 FEED BACK LOOP TEST
:*****
†ST21: SCOPE
      MOV #10,ICOUNT ; ITERATION COUNT
      MOV #21,ERTSTN ; SAVE TEST # FOR ERROR REPCRT
      MOV #SCP21,RETURN ; SCOPE LOOP RETURN ADRS
SCP21:

      ; THE FOLLOWING JUMPERS MUST BE INSERTED--OR USE JUMPER PLUG
      ; PIN 8 TO PIN 2 (-15V TO C OF RELAY)
      ; PIN 5 TO PIN 7 (COLLECTOR TO REMSEN)
      ; PIN 6 TO PIN 3 (EMITER TO N.O. OF RELAY)
      ; PIN 10 TO PIN 3 (INTR.S.T. TO N.O. OF RELAY)
      ; PIN 11 TO PIN 1 THRU A SWITCH (RESTART S.T. TO N.C. OF RELAY)
      ; PIN 9 TO PIN 1 (START S.T. TO N.C. OF RELAY)
      ; PIN 17 TO PIN 16 (CLOCK IN FOR W DENAB)
      ; PIN 15 TO PIN 14 (DATA IN FOR W DENAB)
      ; THIS ROUTINE CHECKS REMOTE SENSE LOGIC, EXTREQ(IN
      ; NON-INTERRUPT MODE), START LOGIC.
:*****
: OPEN SWITCH
:*****
      CLR @WDCSR
      BIT @WDRLY,@WDCSR
      BEQ .+4 ; BRANCH IF NO ERROR CONDITION
      ERROR ; WD RELAY SHOULD NOT BE PICKED
      CMPB @WDCSR,#0
      BEQ .+4 ; BRANCH IF NO ERROR CONDITION
      ERROR ; WDCSR SHOULD BE CLEARED
      BIS @WDENAB,@WDCSR ; ENABLE RELAY
      BIT @WDENAB,@WDCSR
      BNE .+4 ; BRANCH IF NO ERROR CONDITION
      ERROR ; CHECK JUMPERS,WDENAB NOT SET
      CLR @WDDAT
      MOV @WDDAT,(PC)+
STORE: 0 ; PASS CONTENTS OF WDDAT HERE
      BIC #370,@STORE ; CLEAR MNCLK
      CMPB STORE,#4 ; IS WD2 SET ?
      BEQ .+4 ; BRANCH IF NO ERROR CONDITION
      ERROR ; WD2 NOT SET
      BIS @WD1,@WDDAT ; UP DATE SEQ.
      BIT @WD1,@WDDAT
      BNE .+4 ; BRANCH IF NO ERROR CONDITION
      ERROR ; WD1 NOT SET
      BIC @WD2,@WDDAT
      BIT @WD1,@WDDAT
      BEQ .+4 ; BRANCH IF NO ERROR CONDITION
      ERROR ; WD1 NOT CLEARED
; WHEN RELAY PICKS (APPROX. 1 MSEC) IT SETS EXTREQ & REMSEN (20 MSEC LATER)
      JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
      FEEDBK ; FUNCTION TO OCCUR
      JSR R5,HOLDUP ; STALL SUFFICIENT TIME TO ALLOW
      TRIGER ; FUNCTION TO OCCUR
      MOV @WDCSR,(PC)+
STORE1: 0 ; PASS CONTENTS OF WDCSR HERE
    
```

```

1647 005224 126727 177772 000236      CMPB   STORE1,#236
1648 005232 001401                      BEQ    .+4          ;BRANCH IF NO ERROR CONDITION
1649 005234 104000                      ERROR  ;BITS 1,2,3,4,7 NOT SET
1650                                ;FLAG & EXTREQ ARE SET BY INTERRUPT LOGIC
1651                                ;REMSEN SET WHEN RELAY OPENS(SETS)
1652 005236 032777 000002 173744 1$:  BIT    #WDRLY,#WDCSR ;WAIT UNTIL RELAY CLEARS(DROPS)
1653 005244 001374                      BNE   1$
1654 005246 032777 000004 173734      BIT    #WDENAB,#WDCSR
1655 005254 001401                      BEQ    .+4          ;BRANCH IF NO ERROR CONDITION
1656 005256 104000                      ERROR  ;WDENAB NOT CLEARED
1657                                ;WHEN RELAY DROPS (APPROX. 1 MSEC) IT CLEARS WDENAB,
1658                                ; AND APPROX. 20 MSEC LATER WDENAB SETS AGAIN
1659 005260 004567 003034      JSR   RS,HOLDUP    ;STALL SUFFICIENT TIME TO ALLOW
1660 005264 001252                      FEEDBK ;FUNCTION TO OCCUR
1661 005266 032777 000004 173714      BIT    #WDENAB,#WDCSR
1662 005274 001001                      BNE   .+4          ;BRANCH IF NO ERROR CONDITION
1663 005276 104000                      ERROR  ;WDENAB NOT SET
1664                                ;WDENAB SET BY START LOGIC, BUT SHOULD NOT SET WDRLY AGAIN
1665 005300 032777 000002 173702      BIT    #WDRLY,#WDCSR
1666 005306 001401                      BEQ    .+4          ;BRANCH IF NO ERROR CONDITION
1667 005310 104000                      ERROR  ;WDRLY SHOULD NOT BE SET AGAIN
1668 005312 042777 000004 173670      BIC    #WDENAB,#WDCSR ;FORCE CLEAR WDRLY
1669 005320 005027                      CLR   (PC)+        ;WAIT UNTIL LOOP CLEARS
1670 005322 000000      2$:  0
1671 005324 005367 177772      3$:  DEC   2$
1672 005330 001375                      BNE   3$
1673
1674

```

E04

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 44  
DBGAAA.P11 T22 EXTREQ WITH INTERRUPT TEST--FEED BACK LOOP

1675  
1676  
1677  
1678 005332 104400  
1679 005334 012767 000010 005312  
1680 005342 012767 000022 005162  
1681 005350 012767 005356 005302  
1682 005356  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696 005356 005077 173626  
1697 005362 016777 173670 173612  
1698 005370 012777 005460 173602  
1699 005376 016767 173700 172372  
1700 005404 032777 000002 173576  
1701 005412 001401  
1702 005414 104000  
1703 005416 052777 000004 173564  
1704 005424 052777 000100 173556  
1705 005432 052777 000002 173552  
1706 005440 042777 000004 173544  
1707  
1708 005446 005000  
1709 005450 005200  
1710 005452 001376  
1711 005454 104000  
1712 005456 000425  
1713 005460 016777 173516 173512  
1714 005466 012777 000000 173506  
1715 005474 105777 173510  
1716 005500 100401  
1717 005502 104000  
1718 005504 032777 000010 173476  
1719 005512 001001  
1720 005514 104000  
1721 005516 012716 005532  
1722 005522 012767 000340 172246  
1723 005530 000002  
1724 005532 042777 000004 173450  
1725 005540 005027  
1726 005542 000000  
1727 005544 005367 177772  
1728 005550 001375  
1729  
1730

```
*****  
:TEST 22 EXTREQ WITH INTERRUPT TEST--FEED BACK LOOP  
*****  
†ST22: SCOPE  
MOV #10,ICOUNT ;ITERATION COUNT  
MOV #22,ERTSTN ;SAVE TEST # FOR ERROR REPORT  
MOV #SCP22,RETURN ;SCOPE LOOP RETURN ADRS  
SCP22:
```

```
:THE FOLLOWING JUMPERS MUST BE INSERTED--OR USE JUMPER PLUG  
:PIN 8 TO PIN 2 (-15V TO C OF RELAY)  
:PIN 5 TO PIN 7 (COLLECTOR TO REMSEN)  
:PIN 6 TO PIN 3 (EMITER TO N.O. OF RELAY)  
:PIN 10 TO PIN 3 (INTR.S.T. TO N.O. OF RELAY)  
:PIN 11 TO PIN 1, THRU A SWITCH (RESTART S.T. TO N.C. OF RELAY)  
:PIN 9 TO PIN 1 (START S.T. TO N.C. OF RELAY)  
:PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)  
:PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
```

```
*****  
:OPEN SWITCH  
*****
```

```
CLR WDCSR  
MOV WDPRT,WDIS  
MOV #EXTINT,WDIV  
MOV LESS1,PS  
BIT #WDRLY,WDCSR  
BEQ .+4 ;BRANCH IF NO ERROR CONDITION  
ERROR ;WD RELAY SHOULD NOT BE PICKED  
BIS #WDENAB,WDCSR  
BIS #WDIEN,WDCSR  
BIS #WD1,WDDAT ;UP DATE SEQ.  
BIC #WD2,WDDAT  
;STALL UNTIL INTERRUPT OCCURS  
CLR RO  
DELAY1: INC RO ;WAIT UNTIL INTERRUPT  
BNE DELAY1 ;INTERRUPT DID NOT OCCUR  
ERROR  
BR CONTU  
EXTINT: MOV WDIS,WDIV ;RESTORE WDIV  
MOV #0,WDIS  
TSTB WDCSR  
BMI .+4 ;TEST FLAG  
ERROR ;BRANCH IF NO ERROR CONDITION  
;ERROR, INTERRUPT NOT CAUSED BY FLAG  
BIT #EXTREQ,WDCSR  
BNE .+4 ;BRANCH IF NO ERROR CONDITION  
ERROR ;EXTREQ DID NOT GET SET  
MOV #CONTU,(SP) ;LOAD RETURN ADDRESS  
MOV #LEVEL7,PS ;RESTORE PROC. STATUS  
RTI  
CONTU: BIC #WDENAB,WDCSR ;FORCE CLEAR WDRLY  
CLR (PC)+ ;WAIT UNTIL LOOP CLEARS  
25: 0  
35: DEC 25  
BNE 35
```

# F04

MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 45  
 DBQAAA.P11 T23 WD TRIGGER O.S. TRANSISTOR TEST

```

1731 ;*****
1732 ;TEST 23 WD TRIGGER O.S. TRANSISTOR TEST
1733 ;*****
1734 005552 104400 TST23: SCOPE
1735 005554 012767 000010 005072 MOV #10,ICOUNT ;ITERATION COUNT
1736 005562 012767 000023 004742 MOV #23,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
1737 005570 012767 005576 005062 MOV #SCP23,RETURN ;SCOPE LOOP RETURN ADRS
1738 005576 SCP23:
1739
1740 ;THE FOLLOWING JUMPERS MUST BE INSERTED--OR USE JUMPER PLUG
1741 ;PIN 8 TO PIN 2 (-15V TO C OF RELAY)
1742 ;PIN 5 TO PIN 7 (COLLECTOR TO REMSEN)
1743 ;PIN 6 TO PIN 3 (EMITER TO N.O. OF RELAY)
1744 ;PIN 10 TO PIN 3 (INTR.S.T. TO N.O. OF RELAY)
1745 ;PIN 11 TO PIN 1, THRU A SWITCH (RESTART S.T. TO N.C. OF RELAY)
1746 ;PIN 9 TO PIN 1 (START S.T. TO N.C. OF RELAY)
1747 ;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
1748 ;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
1749 ;*****
1750 ; OPEN SWITCH
1751 ;*****
1752 005576 005077 173406 CLR #WDCSR
1753 005602 032777 000002 173400 BIT #WDRLY,#WDCSR
1754 005610 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1755 005612 104000 ERROR ;WDRLY NOT CLEARED--DROPPED
1756 005614 052777 000004 173366 BIS #WDENAB,#WDCSR
1757 005622 052777 000002 173362 BIS #WD1,#WDDAT ;UPDATE SEQ.
1758 005630 042777 000004 173354 BIC #WD2,#WDDAT
1759 ;WAIT UNTIL REMSEN SETS
1760 005636 004567 002456 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1761 005642 001252 FEEDBK ;FUNCTION TO OCCUR
1762 005644 004567 002450 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1763 005650 001272 TRIGER ;FUNCTION TO OCCUR
1764 005652 032777 000020 173330 BIT #REMSEN,#WDCSR
1765 005660 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1766 005662 104000 ERROR ;TRANSISTOR IS OPEN
1767 005664 042777 000004 173316 BIC #WDENAB,#WDCSR ;FORCE CLEAR WDRLY
1768 005672 004567 002422 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1769 005676 001270 TIMEOUT ;FUNCTION TO OCCUR
1770 005700 004567 002414 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1771 005704 001252 FEEDBK ;FUNCTION TO OCCUR
1772 005706 005077 173276 CLR #WDCSR
1773 005712 032777 000002 173270 BIT #WDRLY,#WDCSR
1774 005720 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1775 005722 104000 ERROR ;WDRLY NOT CLEARED--DROPPED
1776 005724 052777 000004 173256 BIS #WDENAB,#WDCSR
1777 005732 012737 010726 000014 MOV #88,#14
1778 005740 012737 000340 000016 MOV #LEVEL7,#16
1779 005746 012746 000360 MOV #360,-(SP) ;SET "T" BIT
1780 005752 012746 005760 MOV #15,-(SP)
1781 005756 000002 RTI ;1ST FALSE RETURN INTERRUPT
1782 ;UP DATE SEQ. FOR 22 MSEC TO OPEN TRANSISTOR,PREVENTING
1783 ;REMSEN SETTING
1784 005760 1S:
1785 005760 004567 002334 JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
1786 005764 001314 TRANST ;FUNCTION TO OCCUR
  
```

G04

.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 46  
 DBQAAA.P11 T23 WD TRIGGER O.S. TRANSISTOR TEST

1787	005766	012746	000340			MOV	#340, -(SP)	;RESTORE STACK
1788	005772	012745	006000			MOV	#CONTT, -(SP)	
1789	005776	000002				RTI		;2ND FALSE RTI STOP UP DATE SEQ
1790	006000	012737	000016	000014	CONTT:	MOV	#16, @#14	;RESTORE TRAP CATCHER
1791	006006	012737	000000	000016		MOV	#0, @#16	
1792	006014	032777	000020	173166		BIT	#REMSEN, @WDCSR	
1793	006022	001401				BEQ	.+4	;BRANCH IF NO ERROR CONDITION
1794	006024	104000				ERROR		;TRANSISTOR DID NOT OPEN
1795								;TRANSISTOR SHOULD BE OPEN PEVENTING REMSEN SETTING
1796	006026	042777	000004	173154		BIC	#WDENAB, @WDCSR	;FORCE CLEAR WDRLY
1797	006034	05027				CLR	(PC)+	;WAIT UNTIL LOOP CLEARS
1798	006036	000000			2S:	0		
1799	006040	005367	177772		3S:	DEC	2S	
1800	006044	001375				BNE	3S	
1801								
1802								

# H04

MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 47  
DBGAAA.P11 T24 END FEED BACK LOOP TEST

```
1803 ;*****  
1804 ;TEST 24 END FEED BACK LOOP TEST  
1805 ;*****  
1806 006046 104400 TST24: SCOPE  
1807 006050 012767 000001 004576 MOV #1,ICOUNT ; ITERATION COUNT  
1808 006056 012767 000024 004446 MOV #24,ERTSTN ; SAVE TEST # FOR ERROR REPORT  
1809 006064 012767 006072 004566 MOV #SCP24,RETURN ; SCOPE LOOP RETURN ADRS  
1810 006072 SCP24:  
1811  
1812 006072 000004 012210 TYPE NEXTST  
1813 006076 000137 007276 JMP #MON12  
1814  
1815
```

```

1816
1817
1818
1819 006102 104400
1820 006104 012767 000001 004542
1821 006112 012767 000025 004412
1822 006120 012767 006126 004532
1823 006126
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837 006126 005077 173056
1838 006132 032777 000002 173050
1839 006140 001401
1840 006142 104000
1841 006144 052777 000004 173036
1842 006152 005037 001312
1843 006156 012737 006240 000004
1844 006164 012737 000340 000006
1845 006172 052777 000002 173012
1846 006200 042777 000004 173004
1847 006206 004567 002106
1848 006212 001252
1849 006214 042777 000004 172766
1850 006222 005027
1851 006224 000000
1852 006226 005367 177772
1853 006232 001375
1854 006234 104000
1855 006236 000420
1856 006240 011667 173046
1857 006244 023737 001332 001312
1858 006252 001401
1859 006254 104000
1860 006256 012737 000006 000004
1861 006264 012737 000000 000006
1862 006272 012716 006300
1863 006276 000002
1864 006300 005027
1865 006302 000000
1866 006304 005367 177772
1867 006310 001375
1868
1869

```

```

;*****
;TEST 25 RESTART TEST
;*****
TST25: SCOPE
MOV #1,ICOUNT ;ITERATION COUNT
MOV #25,ERTSTN ;SAVE TEST # FOR ERROR REPCRT
MOV #SCP25,RETURN ;SCOPE LOOP RETURN ADRS

SCP25:
;THE FOLLOWING JUMPERS MUST BE INSERTED--OR USE JUMPER PLUG
;PIN 8 TO PIN 2 (-15V TO C OF RELAY)
;PIN 5 TO PIN 7 (COLLECTOR TO REMSEN)
;PIN 6 TO PIN 3 (EMMITER TO N.O. OF RELAY)
;PIN 10 TO PIN 3 (INTR.S.T. TO N.O. OF RELAY)
;PIN 11 TO PIN 1, THRU A SWITCH (RESTART S.T. TO N.C. OF RELAY)
;PIN 9 TO PIN 1 (START S.T. TO N.C. OF RELAY)
;PIN 17 TO PIN 16 (CLOCK IN FOR WDENAB)
;PIN 15 TO PIN 14 (DATA IN FOR WDENAB)
;*****
;CLOSE SWITCH
;*****
CLR #WDCSR
BIT #WDRLY,#WDCSR
BEQ .+4 ;BRANCH IF NO ERROR CONDITION
ERROR ;WDRLY NOT CLEARED--DROPPED
BIS #WDENAB,#WDCSR ;ENABLE RELAY
CLR #LOCTRP ;CLEAR TRAP LOCATION FOR RESTART ADDR.
MOV #CRUNCH,#4 ;SET UP SERVICE ROUTINE
MOV #LEVEL7,#6
BIS #WD1,#WDDAT ;UPDATE SEQ.
BIC #WD2,#WDDAT
JSR RS,HOLDUP ;STALL SUFFICIENT TIME TO ALLOW
FEEDBK ;FUNCTION TO OCCUR
BIC #WDENAB,#WDCSR ;FORCE CLEAR WDRLY
CLR (PC)+ ;WAIT UNTIL "CRUNCH" OCCURS

25: 0
35: DEC 25
BNE 35
ERROR ;RESTART DID NOT OCCUR
BR RTNST
CRUNCH: MOV (SP),LOCTRP
CMP #ROMADD,#LOCTRP
BEQ .+4 ;BRANCH IF NO ERROR CONDITION
ERROR ;WRONG RESTART ADDRESS
MOV #6,#4 ;RESTORE TRAPCATCHER
MOV #0,#6
MOV #RTNST,(SP)
RTI
RTNST: CLR (PC)+ ;WAIT UNTIL LOOP CLEARS
45: 0
55: DEC 45
BNE 55

```



J04

```

1870 ;*****
1871 ;TEST 26 END RESTART TEST
1872 ;*****
1873 006312 104400 TST26: SCOPE
1874 006314 012767 000001 004332 MOV #1,ICOUNT ;ITERATION COUNT
1875 006322 012767 000026 004202 MOV #26,ERTSTN ;SAVE TEST # FOR ERROR REPORT
1876 006330 012767 006336 004322 MOV #SCP26,RETURN ;SCOPE LOOP RETURN ADRS
1877 006336 SCP26:
1878
1879 006336 000004 012246 TYPE ENDTST
1880 006342 000137 007276 JMP J#MON12
1881
1882
1883 .SBTTL MONITOR
1884
1885 ;*****
1886 006346 MONITOR:
1887 ;*****
1888
1889
1890
1891 006346 000005 RESET
1892 006350 012706 001100 MOV #BEGIN,SP ;SET UP STACK POINTER
1893 006354 012767 000340 171414 MOV #LEVEL7,PS ;MONITOR AT LEVEL 7
1894 006362 000004 012203 TYPE ,HOME ;HOME UP AND ERASE SCREEN
1895
1896 006366 000004 011054 H: TYPE ,HEADER
1897 006372 012737 006476 006370 MOV #RELOAD,J#H+2 ;HEADER TEXT GETS WIPED BY NPR'S
1898 006400 004737 007340 JSR PC,J#DEFAULT ;SET UP DEFAULT PARAMETERS
1899
1900 MON1.0:
1901 006404 012706 001100 MOV #BEGIN,SP ;SET UP STACK POINTER
1902 006410 000004 011630 TYPE ,FSTART
1903 006414 104004 KEY.TO.R0
1904 006416 122700 000104 CMPB #'D,R0 ;D = DEFAULT PARAMETERS
1905 006422 001003 BNE 1$
1906 006424 004737 007340 JSR PC,J#DEFAULT ;SET UP DEFAULT PARAMETERS
1907 006430 000403 BR 2$
1908 006432 122700 000120 1$: CMPB #'P,R0 ;P = PREVIOUSLY SELECTED PARAMETERS
1909 006436 001002 BNE 3$
1910 006440 000137 007252 JMP J#MON10
1911 006444 122700 000123 3$: CMPB #'S,R0 ;S = GO THROUGH AND SELECT PARAMETERS
1912 006450 001426 BEQ MON1
1913 006452 122700 000116 CMPB #'N,R0 ;N = START AT THIS TEST #
1914 006456 001352 BNE MON1.0
1915 006460 000004 011645 TYPE ,MSGO
1916 006464 104003 .ACCEPTO
1917 006466 016767 004412 172564 MOV OCTNUM,FIRST.TST
1918 006474 000761 BR 2$
1919
1920 006476 051137 046105 040517 RELOAD: .ASCIZ "+RELOAD FOR HEADER TEXT"
1921 006504 020104 047506 020122
1922 006512 042510 042101 051105
1923 006520 052040 054105 000124
1924
1925 .EVEN
;SET UP TEST PARAMETERS

```

```

1926
1927 006526 004737 007340      MON1: JSR    PC,2#DEFAULT ;SET UP DEFAULT PARAMETERS
1928 006532 000004 011645      TYPE    ,MSG0          ;FIRST TEST #
1929
1930 006536 104003              ACCEPTO ;ACCEPT TEST NUMBER FROM KEYBOARD
1931
1932
1933 006540 005767 004340              TST    OCTNUM          ;TEST FOR DEFAULT
1934 006544 001403              BEQ    MON3            ;BRANCH ON DEFAULT
1935 006546 016767 004332 172504      MOV    OCTNUM,FIRST.TST ;LOAD FIRST TEST #
1936
1937 006554 000004 011664      MON3: TYPE    ,MSG1          ;BASE ADDRESS:
1938
1939
1940 006560 104003              ACCEPTO ;ACCEPT BASE ADDRESS FROM KEYBOARD
1941
1942 006562 005767 004316              TST    OCTNUM
1943 006566 001405              BEQ    MON4
1944 006570 016767 004310 172510      MOV    OCTNUM,WDBASE
1945 006576 004767 001774              JSR    PC,WDADDR
1946
1947 006602 000004 011704      MON4: TYPE    ,MSG2          ;ACCEPT INTERRUPT VECTOR
1948 006606 104003              ACCEPTO
1949 006610 005767 004270              TST    OCTNUM
1950 006614 001405              BEQ    MON4.1
1951 006616 016767 004262 172464      MOV    OCTNUM,WDIBAS
1952 006624 004767 002022              JSR    PC,WDEVC
1953
1954
1955 006630 000004 011730      MON4.1: TYPE   ,MSG3          ;PRIORITY
1956 006634 104003              ACCEPTO ;ACCEPT WD PRIORITY LEVEL
1957 006636 005767 004242              TST    OCTNUM          ;TEST FOR DEFAULT
1958 006642 001425              BEQ    MON5            ;BRANCH ON DEFAULT
1959 006644 006367 004234              ASL    OCTNUM
1960 006650 006367 004230              ASL    OCTNUM
1961 006654 006367 004224              ASL    OCTNUM
1962 006660 006367 004220              ASL    OCTNUM
1963 006664 006367 004214              ASL    OCTNUM
1964 006670 016767 004210 172360      MOV    OCTNUM,WDPRT    ;LOAD PRIORITY
1965 006676 005367 004202              DEC    OCTNUM
1966 006702 042737 000037 013104      BIC    #37,2#OCTNUM    ;CLEAR TNZVC
1967 006710 016767 004170 172364      MOV    OCTNUM,LESS1    ;PRIORITY TO ALLOW WD INTERRUPTS
1968
1969
1970
1971 006716 000004 011773      MON5: TYPE    ,MSG5          ;TIME OUT VALUE
1972 006722 004767 001226              JSR    PC,DACCEPT      ;ACCEPT DECIMAL # FROM TTY
1973 006726 005767 172344              TST    DECNUM
1974 006732 001403              BEQ    MON5.1          ;BRANCH ON DEFAULT
1975 006734 016767 172336 172326      MOV    DECNUM,TIMOUT   ;PUT IN NEW VALUE
1976 006742 000004 012043      MON5.1: TYPE   ,MSG6          ;TRIGGER VALUE
1977 006746 004767 001202              JSR    PC,DACCEPT      ;ACCEPT DECIMAL # FROM TTY
1978 006752 005767 172320              TST    DECNUM
1979 006756 001403              BEQ    MON6            ;BRANCH ON DEFAULT
1980 006760 016767 172312 172304      MOV    DECNUM,TRIGER   ;PUT IN NEW VALUE
1981 006766 000004 012123      MON6: TYPE    ,MSG7          ;CLOCKF

```

```

1982 006772 004767 001156      JSR      PC,DACCEP      ;ACCEPT DECIMAL # FROM TTY
1983 006776 005767 172274      TST      DECNUM
1984 007002 001512                   BEQ      MON7          ;BRANCH ON DEFAULT
1985 007004 022767 001750 172264  CYC1K:  CMP      #1000.,DECNUM
1986 007012 001012                   BNE     CYC200
1987 007014 012767 176030 172274  MOV     #-1000.,SEC1    ;SET UP 1000 HZ CONSTANTS
1988 007022 012767 166170 172270  MOV     #-5000.,SEC5   ;
1989 007030 012767 154360 172264  MOV     #-10000.,SEC10 ;
1990 007036 000471                   BR      OUTDEC
1991 007040 022767 000310 172230  CYC200: CMP     #200.,DECNUM
1992 007046 001012                   BNE     CYC100
1993 007050 012767 177470 172240  MOV     #-200.,SEC1    ;SET UP 200 HZ CONSTANTS
1994 007056 012767 176030 172234  MOV     #-1000.,SEC5   ;
1995 007064 012767 174060 172230  MOV     #-2000.,SEC10 ;
1996 007072 000453                   BR      OUTDEC
1997 007074 022767 000144 172174  CYC100: CMP     #100.,DECNUM
1998 007102 001012                   BNE     CYC50
1999 007104 012767 177634 172204  MOV     #-100.,SEC1    ;SET UP 100 HZ CONSTANTS
2000 007112 012767 177014 172200  MOV     #-500.,SEC5   ;
2001 007120 012767 176030 172174  MOV     #-1000.,SEC10 ;
2002 007126 000435                   BR      OUTDEC
2003 007130 022767 000062 172140  CYC50:  CMP     #50.,DECNUM
2004 007136 001012                   BNE     CYC60
2005 007140 012767 177716 172150  MOV     #-50.,SEC1    ;SET UP 50 HZ CONSTANTS
2006 007146 012767 177406 172144  MOV     #-250.,SEC5   ;
2007 007154 012767 177014 172140  MOV     #-500.,SEC10 ;
2008 007162 000417                   BR      OUTDEC
2009 007164 022767 000074 172104  CYC60:  CMP     #60.,DECNUM
2010 007172 001401                   BEQ     .+4            ;BRANCH IF NO ERROR CONDITION
2011 007174 104000                   ERROR    ;UNASSIGNED CLOCK FREQ.
2012 007176 012767 177704 172112  MOV     #-60.,SEC1    ;SET UP 60 HZ CONSTANTS
2013 007204 012767 177324 172106  MOV     #-300.,SEC5   ;
2014 007212 012767 176650 172102  MOV     #-600.,SEC10 ;
2015 007220 000400                   BR      OUTDEC
2016 007222 016767 172050 172044  OUTDEC: MOV     DECNUM,CLOCKF ;PUT IN NEW VALUE
2017 007230 000004 012146      MON7:  TYPE    ,MSG8     ;RESTART ADDRESS
2018 007234 104003                   ACCEPTO ;ACCEPT RESTART ADDRESS
2019 007236 005767 003642      TST     OCTNUM      ;TEST FOR DEFAULT
2020 007242 001403                   BEQ     MON10        ;BRANCH ON DEFAULT
2021 007244 016767 003634 172060  MOV     OCTNUM,ROMADD ;PUT IN NEW VALUE
2022                                     ;ASK FOR DYNAMIC SWITCH SETTINGS ON CONSOL SWITCHES
2023
2024 007252 005067 003252      MON10: CLR     ERRCNT
2025 007256 000004 011755      TYPE    ,MSG4      ;SET DYNAMIC SWITCHES
2026 007262 104004                   KEY.TO.RD ;TYPE ANYTHING
2027 007264 122700 000003      CMPB   #3,RD      ;TEST FOR CONTROL C
2028 007270 001002                   BNE     MON12
2029 007272 000167 177106      JMP     MON10.O    ;RETURN TO MONITOR IF ↑C
2030
2031
2032
2033 007276 016700 171756      MON12: MOV     FIRST.TST,RD ;TEST FOR DEFAULT
2034 007302 001002                   BNE     MON13        ;BRANCH IF NOT DEFAULT
2035 007304 005267 171750      INC     FIRST.TST   ;DEFAULT TEST NUMBER IS ONE
2036 007310 016767 171744 003214  MON13: MOV     FIRST.TST,ERTSTN
2037 007316 006300                   ASL     RD

```

```

2038 007320 016037 012660 012660      MOV      TSTABLE-2(RO),@#RETURN
2039 007326 062737 000024 012660      ADC      #24,@#RETURN
2040 007334 000170 012660      MON14:  JMP      @TSTABLE-2(RO) ;JUMP TO SELECTED TEST
2041
2042
2043                                     ;UTILITIES
2044 007340 012767 000300 171710  DEFAULT: MOV      #LEVEL6,WDPRT
2045 007346 012767 000240 171726      MOV      #LEVEL5,LESS1
2046 007354 012700 001224      MOV      #DFLTAD,RO
2047 007360 004767 000056      JSR      PC,LODAD
2048 007364 012700 001214      MOV      #DFLTIV,RO
2049 007370 004767 000012      JSR      PC,LODIV
2050 007374 012700 001234      MOV      #DFLMSC,RO
2051 007400 004767 000060      JSR      PC,LODMSC
2052 007404 000207      RTS      PC ;SET UP DEFAULT PARAMETERS
2053 007406 012067 171562  LODIV:  MOV      (RO)+,CLKIV
2054 007412 012067 171560      MOV      (RO)+,CLKIS
2055 007416 012067 171556      MOV      (RO)+,WDIV
2056 007422 012067 171554      MOV      (RO)+,WDIS
2057 007426 000207      RTS      PC
2058 007430 012700 001306  FREI:  MOV      #WDBASE,RO
2059 007434 004767 000002      JSR      PC,LODAD
2060 007440 000207      RTS      PC
2061 007442 012067 171536  LODAD:  MOV      (RO)+,CLKCSR
2062 007446 012067 171534      MOV      (RO)+,VOID
2063 007452 012067 171532      MOV      (RO)+,WDCSR
2064 007456 012067 171530      MOV      (RO)+,WDDAT
2065 007462 000207      RTS      PC
2066 007464 012067 171600  LODMSC: MOV      (RO)+,TIMOUT
2067 007470 012067 171576      MOV      (RO)+,TRIGER
2068 007474 012067 171574      MOV      (RO)+,CLOCKF
2069 007500 012067 171626      MOV      (RO)+,ROMADD
2070 007504 012067 171606      MOV      (RO)+,SEC1
2071 007510 012067 171604      MOV      (RO)+,SEC5
2072 007514 012067 171602      MOV      (RO)+,SEC10
2073 007520 000207      RTS      PC
2074                                     ;UTILITIES
2075 007522 013527  TIMEUP: MOV      @ (R5)+,(PC)+ ;FETCH TIMEUP COUNT
2076 007524 000000  TCNT:   MOV      0
2077 007526 004567 000026      JSR      R5,SCALE ;SCALE TO CLOCKF
2078 007532 032777 000200 171444  T.1:   BIT      #FLAG,@CLKCSR
2079 007540 100374      BPL      T.1 ;WAIT FOR TICK
2080 007542 042777 000200 171434      BIC      #FLAG,@CLKCSR ;CLEAR DONE
2081 007550 005367 177750      DEC      TCNT ;KEEP TIME
2082 007554 001366      BNE      T.1 ;BRANCH IF NO TIMEOUT
2083 007556 000205      RTS      R5
2084 007560 022767 001750 171506  SCALE: CMP      #1000.,CLOCKF ;IS CLOCK @ 1KC
2085 007566 001001      BNE      HZ200 ;NO TRY 200 HZ
2086 007570 000541      BR      EXIT ;NO SCALING, EXIT
2087 007572 022767 000310 171474  HZ200: CMP      #200.,CLOCKF ;IS CLOCK @ 200 HZ
2088 007600 001024      BNE      HZ100 ;NO TRY 100 HZ
2089 007602 005067 171476      CLR      XCNT ;CLEAR SCALE LOCATION
2090 007606 026727 177712 000005      CMP      TCNT,#5 ;CAN YOU DIVIDE BY 5
2091 007614 002001      BGE      .+4 ;BRANCH IF NO ERROR CONDITION
2092 007616 104000      ERROR
2093 007620 005267 171460  DIV5:  INC      XCNT ;INCR SCALE LOCATION
    
```

```

2094 007624 162767 000005 177672 SUB #5,TCNT ;DIV. BY 5, SUCCESSIVE SUBTRACTION
2095 007632 005767 177666 TST TCNT ;CAN IT BE SUBTRACTED AGAIN
2096 007636 001401 BEQ OUT1 ;IS IT ZERO
2097 007640 100367 BPL DIV5 ;IS IT POSITIVE
2098 007642 016767 171436 177654 OUT1: MOV XCNT,TCNT ;MOV SCALED VALUE TO TCNT
2099 ;VALUE HAS BEEN SCALED ( 1/5)
2100 007650 000511 BR EXIT
2101 007652 022767 000144 171414 HZ100: CMP #100.,CLOCKF ;IS CLOCK @ 100 HZ
2102 007660 001024 BNE HZ50 ;NO, TRY 50 HZ
2103 007662 005067 171416 CLR XCNT ;CLEAR SCALE LOCATION
2104 007666 026727 177632 000012 CMP TCNT,#12 ;CAN YOU DIVIDE BY 10?
2105 007674 002001 BGE .+4 ;BRANCH IF NO ERROR CONDITION
2106 007676 104000 ERROR ;TRY HIGHER CLOCK FREQ
2107 007700 005267 171400 DIV10: INC XCNT ;INCR SCALE LOCATION
2108 007704 162767 000012 177612 SUB #12,TCNT ;DIVIDE BY 10, SUCCESSIVE SUBTRACTION
2109 007712 005767 177606 TST TCNT ;CAN IT BE SUBTRACTED AGAIN
2110 007716 001401 BEQ OUT2 ;IS IT ZERO
2111 007720 100367 BPL DIV10 ;IS IT POSITIVE
2112 007722 016767 171356 177574 OUT2: MOV XCNT,TCNT ;MOV SCALED VALUE TO TCNT
2113 ;VALUE HAS BEEN SCALED ( 1/10)
2114 007730 000461 BR EXIT
2115 007732 022767 000062 171334 HZ50: CMP #50.,CLOCKF ;IS CLOCK @ 50 HZ
2116 007740 001024 BNE HZ60 ;NO TRY 60 HZ
2117 007742 005067 171336 CLR XCNT ;CLEAR SCALE LOCATION
2118 007746 026727 177552 000024 CMP TCNT,#24 ;CAN YOU DIVIDE BY 20?
2119 007754 002001 BGE .+4 ;BRANCH IF NO ERROR CONDITION
2120 007756 104000 ERROR ;TRY HIGHER CLOCK FREQ
2121 007760 005267 171320 DIV20: INC XCNT ;INCR SCALE LOCATION
2122 007764 162767 000024 177532 SUB #24,TCNT ;DIVIDE BY 20, SUCCESSIVE SUBTRACTION
2123 007772 005767 177526 TST TCNT ;CAN IT BE SUBTRACTED AGAIN
2124 007776 001401 BEQ OUT3 ;IS IT ZERO
2125 010000 100367 BPL DIV20 ;IS IT POSITIVE
2126 010002 016767 171276 177514 OUT3: MOV XCNT,TCNT
2127 ;VALUE HAS BEEN SCALED ( 1/20)
2128 010010 000431 BR EXIT
2129 010012 022767 000074 171254 HZ60: CMP #60.,CLOCKF ;IS CLOCK @ 60 HZ
2130 010020 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
2131 010022 104000 ERROR ;CLOCKF HAS BEEN ASSIGNED AN UNASSIGNED VALUE
2132 010024 005067 171254 CLR XCNT ;CLEAR SCALE LOCATION
2133 010030 026727 177470 000020 CMP TCNT,#20 ;CAN YOU DIVIDE BY 16?
2134 010036 002001 BGE .+4 ;BRANCH IF NO ERROR CONDITION
2135 010040 104000 ERROR ;TRY HIGHER CLOCK FREQ.
2136 010042 005267 171236 DIV16: INC XCNT ;INCR SCALE LOCATION
2137 010046 162767 000020 177450 SUB #20,TCNT ;DIVIDE BY 16, SUCCESSIVE SUBTRACTION
2138 010054 005767 177444 TST TCNT ;CAN IT BE SUBTRACTED AGAIN
2139 010060 001401 BEQ OUT4 ;IS IT ZERO
2140 010062 100367 BPL DIV16 ;IS IT POSITIVE
2141 010064 016767 171214 177432 OUT4: MOV XCNT,TCNT
2142 ;VALUE HAS BEEN SCALED ( 1/16)
2143 010072 000400 BR EXIT
2144 010074 000205 EXIT: RTS RS
2145
2146 ;THIS UTILITY CALCULATES TRIGGER (ONE 20TH OF TIMEOUT)
2147 010076 016767 171166 171174 CTRIG: MOV TIMEOUT,TTRIG ;START CALCULATION OF TRIGER
2148 010104 005067 171174 CLR XCNT
2149 010110 026727 171164 000000 CMP TTRIG,#0 ;CHECK FOR VALID TIMEOUT
    
```

```

2150 010116 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
2151 010120 104000 ERROR ;TIMOUT CANNOT BE ZERO
2152 010122 005267 171156 ONE20TH: INC XCNT
2153 010126 162767 000024 171144 SUB #24,TTRIG ;DIVIDE BY 20, SUCCESSIVE SUBTRACTION
2154 010134 005767 171140 TST TTRIG ;CAN IT BE SUBTRACTED AGAIN
2155 010140 001401 BEQ OUTS ;IS IT ZERO
2156 010142 100367 BPL ONE20TH ;IS IT POSITIVE
2157 010144 016767 171134 171120 OUTS: MOV XCNT,TRIGER ;TRIGER IS NOW ONE 20TH OF TIMOUT
2158 010152 000207 RTS PC
2159
2160
2161 ;ACCEPT DECIMAL NUMBER FROM TTY, UP TO 10000.
2162 010154 000240 DACCEPT: NOP
2163 010156 005067 171114 CLR DECNUM ;CLEAR DECIMAL NUMBER LOCATION
2164 010162 010146 MOV R1,-(SP) ;SAVE R1
2165 010164 010046 MOV R0,-(SP) ;SAVE R0
2166 010166 005001 ACPTD: CLR R1 ;
2167 010170 ACPTD.1:
2168 010170 104004 KEY.TO.R0 ;FETCH AN ASCII CHAR FROM KEYBOARD
2169 010172 120027 000003 CMPB R0,#3 ;CONTROL C?
2170 010176 001002 BNE AD.2
2171 010200 000137 006404 JMP @MONI.0
2172 010204 122700 000177 AD.2: CMPB #177,R0 ;TEST FOR RUBOUT
2173 010210 001433 BEQ RUBOUD
2174 010212 122700 000015 CMPB #15,R0 ;TEST FOR <CR>
2175 010216 001433 BEQ CARGD
2176 010220 120027 000040 CMPB R0,#40 ;EXIT IF SPACE
2177 010224 001430 BEQ CARGD
2178 010226 120027 000000 CMPB R0,#0 ;TEST FOR VALID DECIMAL NUMBER
2179 010232 002422 ELT RUBOUD
2180 010234 120027 000071 CMPB R0,#9
2181 010240 003017 BGT RUBOUD
2182 010242 042700 177760 BIC #177760,R0 ;CONVERT ASCII TO DECIMAL
2183 010246 006301 ASL R1
2184 010250 010167 171022 MOV R1,DECNUM
2185 010254 006301 ASL R1
2186 010256 006301 ASL R1
2187 010260 066701 171012 ADD DECNUM,R1
2188 010264 060001 ADD R0,R1 ;CHALK'M UP
2189 010266 022701 023420 CMP #10000.,R1
2190 010272 100001 BPL .+4 ;BRANCH IF NO ERROR CONDITION
2191 010274 104000 ERROR ;NUMBER EXCEEDS 10000.
2192 010276 000734 BR ACPTD.1 ;FETCH NEXT CHAR
2193 010300 000004 012304 RUBOUD: TYPE .QUES ;TYPE?
2194 010304 000730 BR ACPTD
2195 010306 010167 170764 CARGD: MOV R1,DECNUM ;PLACE DECIMAL NUMBER HERE
2196 010312 012600 MOV (SP)+,R0 ;RESTORE R0
2197 010314 012601 MOV (SP)+,R1 ;RESTORE R1
2198 010316 000207 RTS PC
2199
2200
2201 ;STALL FOR "X" MILLISECS.
2202 010320 013527 HOLDUP: MOV @R5+,(PC)+
2203 010322 000000 HOLDIT: 0 ;MOVE PASSED ARG. INTO HERE
2204 010324 004767 000200 JSR PC,SCALEF
2205 010330 062767 000001 177764 ADD #1,HOLDIT
    
```

```

2206 010336 032777 000010 170646 BIT #MNCLK, @WDDAT
2207 010344 001014 BNE SVCHH ;START COUNTING @ 1 TRANSITIONS
2208 010346 032777 000010 170636 SVCLL: BIT #MNCLK, @WDDAT ;CHECK FOR 1 STATE
2209 010354 001374 BNE SVCLL ;LOOP UNTIL MNCLK=0
2210 010356 005367 177740 DEC HOLDIT ;@ EVERY 0 TRANSITION
2211 010362 001421 BEQ OUTMN
2212 010364 032777 000010 170620 SVCHL: BIT #MNCLK, @WDDAT ;CHECK FOR 0 STATE
2213 010372 001774 BEQ SVCHL ;LOOP UNTIL MNCLK=1
2214 010374 000764 BR SVCLL
2215 010376 032777 000010 170606 SVCHH: BIT #MNCLK, @WDDAT ;CHECK FOR 0 STATE
2216 010404 001774 BEQ SVCHH ;LOOP UNTIL MNCLK=1
2217 010406 005367 177710 DEC HOLDIT ;@ EVERY 1 TRANSITION
2218 010412 001405 BEQ OUTMN
2219 010414 032777 000010 170570 SVCLH: BIT #MNCLK, @WDDAT ;CHECK FOR 1 STATE
2220 010422 001374 BNE SVCLH ;LOOP UNTIL MNCLK=0
2221 010424 000764 BR SVCHH
2222 010426 000205 OUTMN: RTS RS
2223
2224 ;THIS UTILITY SCALES HOLDIT BY 4/5
2225 010430 022767 000074 170636 SCALEN: CMP #60.,CLOCKF ;IS IT 60 HZ?
2226 010436 001033 BNE OUT6
2227 010440 005067 170640 CLR XCNT
2228 010444 026727 177652 000000 CMP HOLDIT, #0 ;CHECK FOR VALID NUMBER
2229 010452 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
2230 010454 104000 ERROR ;HOLDIT CANNOT BE ZERO
2231 010456 026727 177640 000005 CMP HOLDIT, #5. ;IS IT LESS THAN 5?
2232 010464 100420 BMI OUT6 ;YES, LEAVE HOLDIT AS IS
2233 010466 006367 177630 MULT4: ASL HOLDIT ;MULTIPLY BY 2
2234 010472 006367 177624 ASL HOLDIT ;MULTIPLY BY 2
2235 010476 005267 170602 ONE5TH: INC XCNT
2236 010502 162767 000005 177612 SUB #5, HOLDIT ;DIVIDE BY 5, SUCCESSIVE SUBTRACTION
2237 010510 005767 177606 TST HOLDIT ;CAN IT BE SUBTRACTED AGAIN?
2238 010514 001401 BEQ OUT6.1 ;NO
2239 010516 100367 BPL ONE5TH ;YES
2240 010520 016767 170560 177574 OUT6.1: MOV XCNT, HOLDIT ;HOLDIT IS NOW APPROX. 4/5 OF ORIGINAL
2241 010526 000207 OUT6: RTS PC
2242
2243 ;THIS UTILITY SCALES HOLDIT BY 5/4
2244 010530 022767 000074 170536 SCALEF: CMP #60.,CLOCKF ;IS IT 60 HZ ?
2245 010536 001016 BNE OUT7 ;NO
2246 010540 016767 177556 170536 MOV HOLDIT, XCNT ;SAVE HOLDIT
2247 010546 006367 177550 MULT5: ASL HOLDIT
2248 010552 006367 177544 ASL HOLDIT
2249 010556 066767 170522 177536 ADD XCNT, HOLDIT
2250 010564 006267 177532 ONE4TH: ASR HOLDIT
2251 010570 006267 177526 ASR HOLDIT
2252 010574 000207 OUT7: RTS PC
2253
2254
2255 ;NEW WATCH DOG ADDRESSES
2256 010576 016767 170504 170400 WADDR: MOV WDBASE, CLKCSR
2257 010604 062737 000002 001306 ADD #2, @WDBASE
2258 010612 013767 001306 170366 MOV @WDBASE, VOID
2259 010620 062737 000002 001306 ADD #2, @WDBASE
2260 010626 013767 001306 170354 MOV @WDBASE, WDCSR
2261 010634 062737 000002 001306 ADD #2, @WDBASE

```

```

2262 010642 013767 001306 170342      MOV      @#WDBASE,WDDAT
2263 010650 000207                      RTS PC
2264
2265
2266
2267 010652 016767 170432 170314  WVEC:  :NEW WATCH DOG VECTORS
2268 010660 062737 000002 001310      MOV      WDIBAS,CLKIV
2269 010666 013767 001310 170302      ADD      #2,@#WDIBAS
2270 010674 062737 000002 001310      MOV      @#WDIBAS,CLKIS
2271 010702 013767 001310 170270      ADD      #2,@#WDIBAS
2272 010710 062737 000002 001310      MOV      @#WDIBAS,WDIV
2273 010716 013767 001310 170256      ADD      #2,@#WDIBAS
2274 010724 000207                      MOV      @#WDIBAS,WDIS
2275 010726 052777 000002 170256  BB:    RTS      PC
2276 010734 042777 000004 170250      BIS      #WD1,@WDDAT
2277 010742 000006                      BIC      #WD2,@WDDAT
2278
2279 010744 012777 010776 170222      RTT
2280 010752 016767 170324 167016      :THIS ROUTINE RINGS BELL AT INTERVAL DETERMINED BY RD
2281 010760 016777 170272 170210      TIME:  MOV      #TIME2,@CLKIV      ;SET UP VECTOR RETURN
2282 010766 052777 000100 170210      MOV      LESS1,PS      ;SET PROC. PRIORITY
2283 010774 000777                      MOV      WDPRT,@CLKIS
2284 010776 005200                      TIME1:  BIS      #100,@CLKCSR      ;ENABLE INTERRUPT
2285 011000 001020                      BR       .              ;WAIT,,AFTER SERVICING INTERRUPT "RETURN HERE"
2286 011002 042777 000100 170174      TIME2:  INC      RD
2287 011010 000004 012206                      BNE     TIME3
2288 011014 012716 011052                      BIC     #CLKEN,@CLKCSR      ;CLEAR INTERRUPT ENABLE
2289 011020 012766 000340 000002      TYPE   BELL      ;RING BELL
2290 011026 016777 170144 170140      MOV     #TIME4,(SP)      ;CLEAR OUT "RETURN HERE"
2291 011034 012777 000000 170134      MOV     #LEVEL7,2(SP)
2292 011042 042777 000200 170134      MOV     CLKIS,@CLKIV      ;RESTORE TRAPCATCHER
2293 011050 000002                      MOV     #0,@CLKIS
2294 011052 000207                      TIME3:  BIC     #TICK,@CLKCSR      ;CLEAR TICK
2295
2296
2297
2298 011054 057537 040515 047111      TIME4:  RTS      PC      ;RETURN TO PROGRAM
2299 011062 042504 026503 030461      .SBTTL  MESSAGES
2300 011070 042055 050502 040501      LF=12
2301 011076 040455 042055                      CR=15
2302
2303 011102 057537 054504 040516      HEADER: .ASCII  "+MAINDEC-11-DBQAA-A-D"
2304 011110 044515 020103 053523
2305 011116 052111 044103 051440
2306 011124 052105 044524 043516
2307 011132 173
2308 011133 137 020040 020040      .ASCII  "+ SR 15 HALT ON ERROR"
2309 011140 020040 051440 020122
2310 011146 032461 020040 020040
2311 011154 040510 052114 047440
2312 011162 020116 051105 047522
2313 011170 122
2314 011171 137 020040 020040      .ASCII  "+ SR 14 SCOPE LOOP"
2315 011176 020040 051440 020122
2316 011204 032061 020040 020040
2317 011212 041523 050117 020105

```



2318	011220	047514	050117			
2319	011224	020137	020040	020040	.ASCII "	SR 13 INHIBIT ERROR PRINTOUT"
2320	011232	020040	051123	030440		
2321	011240	020063	020040	044440		
2322	011246	044116	041111	052111		
2323	011254	042440	051122	051117		
2324	011262	050040	044522	052116		
2325	011270	052517	124			
2326	011273	137	020040	020040	.ASCII "	SR 12 SHORT ERROR REPORT"
2327	011300	020040	051440	020122		
2328	011306	031061	020040	020040		
2329	011314	044123	051117	020124		
2330	011322	051105	047522	020122		
2331	011330	042522	047520	052122		
2332	011336	020137	020040	020040	.ASCII "	SR 11 INHIBIT ITERATIONS"
2333	011344	020040	051123	030440		
2334	011352	020061	020040	044440		
2335	011360	044116	041111	052111		
2336	011366	044440	042524	040522		
2337	011374	044524	047117	123		
2338	011401	137	052137	050131	.ASCII "++TYPE: <D>,FOR DEFAULT PARAMETERS"	
2339	011406	035105	020040	036040		
2340	011414	037104	043054	051117		
2341	011422	042040	043105	052501		
2342	011430	052114	050040	051101		
2343	011436	046501	052105	051105		
2344	011444	123				
2345	011445	137	020040	020040	.ASCII "	<P>,FOR PREVIOUS PARAMETERS"
2346	011452	020040	020040	050074		
2347	011460	026076	047506	020122		
2348	011466	051120	053105	047511		
2349	011474	051525	050040	051101		
2350	011502	046501	052105	051105		
2351	011510	123				
2352	011511	137	020040	020040	.ASCII "	<S>,FOR SELECT PARAMETERS"
2353	011516	020040	020040	051474		
2354	011524	026076	047506	020122		
2355	011532	042523	042514	052103		
2356	011540	050040	051101	046501		
2357	011546	052105	051105	123		
2358	011553	137	020040	020040	.ASCIZ "	<N>,FOR START WITH THIS TEST NUMBER"
2359	011560	020040	020040	047074		
2360	011566	026076	047506	020122		
2361	011574	052123	051101	020124		
2362	011602	044527	044124	052040		
2363	011610	044510	020123	042524		
2364	011616	052123	047040	046525		
2365	011624	042502	000122			
2366	011630	057537	026104	026120	FSTART: .ASCIZ "++D,P,S,N,?"	
2367	011636	026123	026116	020077		
2368	011644	000				
2369	011645	137	042524	052123	MSG0: .ASCIZ "++TEST NUMBER: "	
2370	011652	047040	046525	042502		
2371	011660	035122	000040			
2372	011664	041137	051501	020105	MSG1: .ASCIZ "++BASE ADDRESS: "	
2373	011672	042101	051104	051505		

2374	011700	035123	000040			
2375	011704	044537	052116	051105	MSG2:	.ASCIZ "+INTERRUPT VECTOR: "
2376	011712	052522	052120	053040		
2377	011720	041505	047524	035122		
2378	011726	000040				
2379	011730	053537	020104	051120	MSG3:	.ASCIZ "+WD PRIORITY LEVEL: "
2380	011736	047511	044522	054524		
2381	011744	046040	053105	046105		
2382	011752	020072	000			
2383	011755	137	042523	020124	MSG4:	.ASCIZ "+SET SWITCHES"
2384	011762	053523	052111	044103		
2385	011770	051505	000			
2386	011773	137	044524	042515	MSG5:	.ASCIZ "+TIME OUT VALUE IN MILLISECS+TOLERANCE:"
2387	012000	047440	052125	053040		
2388	012006	046101	042525	044440		
2389	012014	020116	044515	046114		
2390	012022	051511	041505	025523		
2391	012030	047524	047514	040522		
2392	012036	041516	035105	000		
2393	012043	137	051124	043511	MSG6:	.ASCIZ "+TRIGER VALUE IN MILLISECS(5% OF TIMOUT VALUE):"
2394	012050	051105	053040	046101		
2395	012056	042525	044440	020116		
2396	012064	044515	046114	051511		
2397	012072	041505	024123	022465		
2398	012100	047440	020106	044524		
2399	012106	047515	052125	053040		
2400	012114	046101	042525	035051		
2401	012122	000				
2402	012123	137	046103	041517	MSG7:	.ASCIZ "+CLOCK FREQ IN HZ:"
2403	012130	020113	051106	050505		
2404	012136	044440	020116	055110		
2405	012144	000072				
2406	012146	051137	051505	040524	MSG8:	.ASCIZ "+RESTART ADDRESS:"
2407	012154	052122	040440	042104		
2408	012162	042522	051523	000072		
2409	012170	042537	051122	051117	ERPC:	.ASCIZ "+ERROR PC "
2410	012176	050040	020103	000		
2411	012203	035	000037		HOME:	.ASCIZ <35><37>
2412	012206	000207			BELL:	.ASCIZ <207>
2413	012210	042137	020117	042522	NEXTST:	.ASCIZ "+DO RESTART TEST,CLOSE SWITCH"
2414	012216	052123	051101	020124		
2415	012224	042524	052123	041454		
2416	012232	047514	042523	051440		
2417	012240	044527	041524	000110		
2418	012246	042537	042116	052040	ENDTST:	.ASCIZ "+END TEST"
2419	012254	051505	000124			
2420	012260	020137	051105	047522	ECM:	.ASCIZ "+ ERRORS DETECTED: "
2421	012266	051522	042040	052105		
2422	012274	041505	042524	035104		
2423	012302	000040				
2424	012304	006477	000012		.QUES:	.ASCIZ "?<CR><LF>
2425	012310	020040	020040	000	SPAC4:	.ASCIZ <40><40><40><40>
2426	012315	040	000		SPACE:	.ASCIZ <40>
2427	012317	015	000012		CRLF:	.ASCIZ <CR><LF>
2428					.EVEN	
2429					.SBTTL	MONITOR FILES

```

2430
2431 ;EMULATOR DECODER ROUTINE
2432
2433 EMTDECODER:
2434   MOV      @R6, -(R6)      ; DUPLICATE PC ON STACK
2435   SUB      #2, @R6        ; POINT PC TO EMT INST.
2436   MOV      @R6, @R6       ; MOV EMT INST ONTO STACK
2437   CMPB    @R6, #20       ; TEST THAT CALL IS WITHIN LIMITS
2438   BLOS    EMTOK          ; BRANCH IF WITHIN LIMITS
2439   ERROR
2440
2441 EMTOK:   ROL      @R6      ; EMT ARGUMENT X 2.
2442   BIC      #177001, @R6   ; CLEAR HIGH BYTE
2443   ADD      @EMTAG, @R6    ; FORM ADRS OF ROUTINE ADRS
2444   MOV      @R6, @R6       ; PUT ROUTINE ADRS ON STACK
2445   JMP      @R6            ; JUMP TO ROUTINE
2446
2447 EMTAG:
2448   .BLKW   20.            ; BEGINNING OF EMT TABLE
2449
2450 ; RESERVE 16. WORDS FOR ADRS LIST
2451
2452 HLT SW=BIT15
2453 LOP SW=BIT14
2454 T.ERROR:
2455
2456 ETMPO:  TYPE      ERPC
2457   MOV      @SP, (PC)+    ; SAVE ERROR PC +2
2458   SUB      #2, ETMPO     ; HERE
2459   MOV      TTY, -(SP)    ; CORRECT PC
2460   MOV      @ETMPO, TTY   ; SAVE TTY
2461   JSR      PC, PRINTR    ; TYPE IN OCTAL
2462   MOV      (SP)+, TTY    ; TYPR LEADING ZERO'S
2463   BIT      #HLT SW, SR   ; RESTORE TTY
2464   BEQ      ERRLOP       ; TEST FOR HALT ON ERROR
2465   HALT
2466
2467 PERRPC: BIT      #HLT SW, SR ; BRANCH IF NO HALT
2468   BEQ      ERRLOP       ; HALT ON ERROR
2469
2470 ERRLOP: BIT      #LOPSW, SR ; TEST FOR SCOPE LOOP
2471   BEQ      EXTR1
2472   MOV      #BEGIN, %6    ; REINIT STACK POINTER
2473   MOV      LESS1, %5     ; WD PRIORITY MINUS ONE
2474   JMP      @RETURN
2475
2476 EXTR1:  RTI
2477
2478 ERRCNT: 0
2479
2480 ERTSTN: 0
2481
2482 ERFLG:  0 ; ERROR CONTROL FLAG -1=MAP ERROR
2483
2484 ; SCOPE LOOP AND CONTROL SUBROUTINE
2485
2486 SCOPEC: TSTB     @TKS
2487   BPL     SCOPEH
2488   MOV     @TKB, (PC)+
2489
2490 DTMP:   0
2491   BIC     #200, DTMP
2492   CMPB   DTMP, #3
2493   BNE    SCOPEH
2494   JMP    @MON1.0
2495
2496 SCOPEH: BIT      #BIT14, SR ; TEST FOR SCOPE
2497   BNE    SCOPEB          ; BRANCH IF SCOPE SELECTED

```

```

2486 012604 032767 004000 164756      BIT      #BIT11,SR      ;TEST FOR ITERATIONS
2487 012612 001015                      BNE      SCOPEA   ;EXIT IF ITERATIONS INHIBITED
2488
2489
2490
2491
2492
2493 012614 026767 000036 000032      CMP      SCOPEF,ICOUNT ;TEST FOR COMPLETION OF ITERATIONS
2494 012622 001407                      BEQ      SCOPEG   ;BRANCH IF COMPLETE
2495 012624 005267 000026                      INC      SCOPEF   ;INCREMENT ITERATION COUNT
2496 012630 005726                      SCOPEB: TST      (SP)+ ;POP RETURN PC
2497 012632 012667 165140                      MOV      (SP)+,PS ;RESTOR PROCESSOR STATUS
2498 012636 000177 000016                      JMP      @RETURN  ;
2499 012642 011667 000012                      SCOPEG: MOV      @SP,RETURN ;SET UP SCOPE RETURN ADRS
2500 012646 005067 000004                      SCOPEA: CLR      SCOPEF   ;CLEAR ITERATION COUNT
2501 012652 000002
2502
2503 012654 000001                      ICOUNT: 1          ;NUMBER OF REQUESTED ITERATIONS
2504 012656 000000                      SCOPEF: 0          ;ITERATION COUNT
2505 012660 001334                      RETURN: TST1      ;DEFAULT RETURN
2506 012662                      TSTABLE:          ;BEGINNING OF TABLE OF TEST ADDRESSES
2507                      .=. +100         ;TEST ADDRESS LIST
2508
2509
2510                      ;ACCEPT OCTAL NUMBER FROM TTY
2511 012762                      T.ACCEPTO:
2512 012762 005067 000116                      CLR      OCTNUM   ;CLEAR OCTAL NUMBER LOCATION
2513 012766 010146                      MOV      R1,-(SP) ;SAVE R1
2514 012770 010046                      MOV      R0,-(SP) ;SAVE R0
2515 012772 005001                      ACPTO:  CLR      R1 ;
2516 012774 104004                      ACPTO.1: KEY.TO.R0 ;FETCH AN ASCII CHAR FROM KEYBOARD
2517 012776 120027 000003                      CMPB    R0,#3     ;CONTROL C?
2518 013002 001002                      BNE     AO.2
2519 013004 000137 006404                      JMP     @MONI.0
2520 013010 122700 000177                      AO.2:  CMPB    #177,R0 ;TEST FOR RUBOUT
2521 013014 001423                      BEQ     RUBOUT
2522 013016 122700 000015                      CMPB    #15,R0   ;TEST FOR <CR>
2523 013022 001423                      BEQ     CARG
2524 013024 120027 000040                      CMPB    R0,#40   ;EXIT IF SPACE
2525 013030 001420                      BEQ     CARG
2526 013032 120027 000000                      CMPB    R0,#0    ;TEST FOR VALID OCTAL NUMBER
2527 013036 002412                      BLT     RUBOUT
2528 013040 120027 000067                      CMPB    R0,#'7
2529 013044 003007                      BGT     RUBOUT
2530 013046 042700 177770                      BIC     #177770,R0 ;CONVERT ASCII TO OCTAL
2531 013052 006301                      ASL     R1
2532 013054 006301                      ASL     R1
2533 013056 006301                      ASL     R1
2534 013060 050001                      BIS     R0,R1    ;CHALK'M UP
2535 013062 000744                      BR      ACPTO.1  ;FETCH NEXT CHAR
2536
2537 013064 000004 012304                      RUBOUT: TYPE      .QUES ;TYPE?
2538 013070 000740                      BR      ACPTO
2539 013072 010167 000006                      CARG:  MOV      R1,OCTNUM ;PLACE OCTAL NUMBER HERE
2540 013076 012600                      MOV     (SP)+,R0 ;RESTORE R0
2541 013100 012601                      MOV     (SP)+,R1 ;RESTOR R1
2541 013102 000002                      RTI
  
```

2542	013104	000000			OCTNUM: 0		
2543					;FETCH AN ASCII CHARACTER FROM KEYBOARD		
2544							
2545	013106				T.KEY.TO.RD:		
2546	013106	105777	165272		TSTB @TKS	;TEST FOR DONE	
2547	013112	100375			BPL -4	;WAIT FOR KEYBOARD	
2548	013114	117700	165266		MOVB @TKB,RO	;FETCH CHAR	
2549	013120	117777	165262	165264	MOVB @TKB,@TPB	;ECHO	
2550	013126	004767	000006		JSR PC,TTYFLG	;WAIT FOR DONE	
2551	013132	042700	177600		BIC #177600,RO	;7 BIT ASCII	
2552	013136	000002			RTI		
2553					;TEST FOR TRANSMITTER DONE		
2554							
2555	013140				TTYFLG:		
2556	013140	105777	165244		TSTB @TPS		
2557	013144	100375			BPL TTYFLG		
2558	013146	000207			RTS PC		
2559							
2560	013150	010567	000112		.IOT: MOV TTY, SAV	;SAVE TTY	
2561	013154	032767	020000	164406	BIT #BIT13,SR	;TEST FOR INHIBIT PRINT	
2562	013162	001032			BNE .TERM		
2563	013164	017605	000000		MOV @ (6), TTY	;GET ADDRESS TO BE TYPED	
2564	013170	122715	000044		.MORE: CMPB #'S, (TTY)	;TERMINATOR?	
2565	013174	001425			BEQ .TERM		
2566	013176	105715			TSTB (TTY)	;TERMINATOR?	
2567	013200	001423			BEQ .TERM		
2568	013202	122715	000001		CMPB #1, (TTY)	;RESTORE OLD SEQUENCE	
2569	013206	001416			BEQ .REST		
2570	013210	122715	000137		CMPB #'C, (TTY)	;SET UP CR LF	
2571	013214	001406			BEQ .CRLF		
2572	013216	105777	165166		TSTB @TPS		
2573	013222	100375			BPL -4		
2574	013224	112577	165162		MOVB (TTY)+, @TPB		
2575	013230	000757			BR .MORE		
2576	013232	005205			.CRLF: INC TTY		
2577	013234	010546			MOV TTY, -(6)		
2578	013236	012705	013270		MOV #.CAR, TTY		
2579	013242	000752			BR .MORE		
2580	013244	012605			.REST: MOV (6)+, TTY		
2581	013246	000750			BR .MORE		
2582	013250	062716	000002		.TERM: ADD #2, (6)		
2583	013254	016705	000006		MOV .SAV, TTY		
2584	013260	004767	177654		.IOTE: JSR PC, TTYFLG	;WAIT FOR DONE	
2585	013264	000002			RTI		
2586							
2587	013266	000000			.SAV: 0		
2588	013270	005015	001		.CAR: .ASCII <CR><LF><1>		
2589		013274			.EVEN		
2590	013274	000000			.TYPE: 0		
2591							
2592	013276	000000	000000	000000	PRINT2: .WORD 0,0,0,0		
2593	013304	000000					
2594	013306	000	000		PRINT3: .BYTE 0,0		
2595							
2596	013310	112767	000001	177770	PRINTR: MOVB #1, PRINT3	;SET ZERO FILL SWITCH	
2597	013316	000402			BR .+6		

2598	013320	005067	177762		PRINTS: CLR	PRINT3		;SUPRESS LEADING ZERO'S
2599	013324	112767	177772	177755	MOV	#-6,PRINT3+1		;SET COUNT
2600	013332	032767	020000	164230	BIT	#BIT13,SR		
2601	013340	001041			BNE	PRTE		
2602	013342	010446			MOV	%4,-(6)		;SAVE R4
2603	013344	012704	013276		MOV	#PRINT2,%4		;SET POINTER TO FIRST ASCII CHAR.
2604	013350	105014			CLRB	(4)		;CLEAR FIRST BYTE
2605	013352	000405			BR	PRINTF		;ROTATE FORST BIT
2606	013354	105014			PRINTL: CLRB	(4)		;CLEAR BYTE OF CHARACTER
2607	013356	006105			ROL	TTY		;ROTATE BIT INTO C
2608	013360	106114			ROLB	(4)		;PACK IT
2609	013362	006105			ROL	TTY		;ROTATE BIT INTO C
2610	013364	106114			ROLB	(4)		;PACK IT
2611	013366	006105			PRINTF: ROL	TTY		;ROTATE BIT INTO C
2612	013370	106114			ROLB	(4)		;PACK IT
2613	013372	105714			TSTB	(4)		
2614	013374	001402			BEQ	+.6		
2615	013376	105267	177704		INCB	PRINT3		
2616	013402	105767	177700		TSTB	PRINT3		;CHECK FILL SWITCH
2617	013406	001402			BEQ	+.6		
2618	013410	152724	000060		BISB	#'0,(4)+		;MAKE INTO ASCII CHAR
2619	013414	105267	177667		INCB	PRINT3+1		
2620	013420	001355			BNE	PRINTL		;REPEAT
2621	013422	022704	013276		CMP	#PRINT2,%4		
2622	013426	001002			BNE	+.6		
2623	013430	112724	000060		MOV	#'0,(4)+		
2624	013434	105014			CLRB	(4)		
2625	013436	000004	013276		TYPE	PRINT2		;TYPE IT
2626	013442	012604			MOV	(6)+,%4		;RESTORE R5
2627	013444	000207			PRTE: RTS	%7		
2628					;SAVE REGS 0 TO 4 SUBROUTINE,			
2629	013446	012667	000032		↑.SAVRG: MOV	(6)+,SVRPC		;SAVE PC AND PSW.
2630	013452	012667	000030		MOV	(6)+,SVRPSW		
2631	013456	010546			MOV	%5,-(6)		
2632	013460	010446			MOV	%4,-(6)		;SAVE REGS 0 - 4
2633	013462	010346			MOV	%3,-(6)		;IN STACK.
2634	013464	010246			MOV	%2,-(6)		
2635	013466	010146			MOV	%1,-(6)		
2636	013470	010046			MOV	%0,-(6)		
2637	013472	016746	000010		MOV	SVRPSW,-(6)		;RESTORE PC AND PSW.
2638	013476	016746	000002		MOV	SVRPC,-(6)		
2639	013502	000002			RTI			;EXIT.
2640	013504	000000			SVRPC: 0			
2641	013506	000000			SVRPSW: 0			
2642					;RESTORE REGS 0 TO 4 SUBROUTINE.			
2643	013510	012667	000032		↑.RSTRG: MOV	(6)+,RSTPC		;SAVE PC AND PSW.
2644	013514	012667	000030		MOV	(6)+,RSTPSW		
2645	013520	012600			MOV	(6)+,%0		;RESTORE REGS 0 - 4
2646	013522	012601			MOV	(6)+,%1		;FROM STACK.
2647	013524	012602			MOV	(6)+,%2		
2648	013526	012603			MOV	(6)+,%3		
2649	013530	012604			MOV	(6)+,%4		
2650	013532	012605			MOV	(6)+,%5		
2651								
2652	013534	016746	000010		MOV	RSTPSW,-(6)		;RESTORE PC AND PSW.
2653	013540	016746	000002		MOV	RSTPC,-(6)		

```

2654
2655 013544 000002          RTI          ;EXIT
2656 013546 000000          RSTPC:  0
2657 013550 000000          RSTPSW: 0
2658
2659          ;POWER FAIL ROUTINE
2660          ;IF SELECTED VERIFY STATUS-IN IS UP
2661          ;AND CE AND DE ARE PRESENTED AS STATUS
2662
2663 013552 104001          PFAIL:  SAVRG
2664 013554 010667 000054          MOV      R6,SAVR6
2665 013560 012767 013570 164236          MOV      #PWRUP,24
2666 013566 000000          HALT
2667          ;POWER UP ROUTINE
2668 013570 000240          PWRUP:  NOP          ;PATCH ANYONE?
2669 013572 016706 000036          MOV      SAVR6,R6
2670 013576 104002          RSTRG
2671 013600 012767 013552 164216          MOV      #PFAIL,24          ;RESTORE POWER FAIL VECTOR
2672 013606 005027          CLR      (PC)+
2673 013610 000000          0
2674 013612 005367 177772          DEC      .-2
2675 013616 001375          BNE      .-4
2676 013620 000004 013636          TYPE      PFLD
2677 013624 012667 164146          MOV      (SP)+,PS
2678 013630 000177 177024          JMP      @RETURN
2679 013634 000000          SAVRE:  0
2680 013636 050137 053517 051105          PFLD:  .ASCIZ  "+POWER FAILED+"
2681 013644 043040 044501 042514
2682 013652 057504 000
2683          .EVEN
2684
2685          000001          .END
  
```







HZ200	007572	2085	2087#											
HZ50	007732	2102	2115#											
HZ60	010012	2116	2129#											
ICOUNT	012654	921*	968*	1001*	1037*	1057*	1115*	1188*	1226*	1254*	1276*	1314*	1367*	1409*
INTTST	004162	1468*	1510*	1564*	1595*	1679*	1735*	1807*	1820*	1874*	2493	2503#		
KEY.T0=	104004	1423	1440#											
LESS1	001302	803#	1903	2026	2168	2515								
LEVEL0=	000000	901#	975	1012	1025	1121	1140	1424	1699	1967*	2045*	2280	2468	
LEVEL1=	000040	777#												
LEVEL2=	000100	778#												
LEVEL3=	000140	779#												
LEVEL4=	000200	780#												
LEVEL5=	000240	781#												
LEVEL6=	000300	782#	2045											
LEVEL7=	000340	783#	891	2044										
LF =	000012	784#	809	812	815	818	826	838	946	992	1018	1067	1160	1453
LKTEST	001360	1722	1778	1844	1893	2289								
LK1	001372	795#	2296#	2424	2427	2588								
LK10	002752	927#												
LK10A	003042	931#												
LK10B	003036	1196#												
LK11	003114	1212#	1216											
LK12	003206	1211#	1219											
LK12A	003224	1232#												
LK2	001402	1260#												
LK2A	001410	1264#	1269											
LK3	001424	935#												
LK4	001464	937#	940											
LK5	001554	938	943#											
LK5A	001620	952#												
LK5B	001630	974#												
LK5.1	001670	982#	983											
LK6	001716	976	986#											
LK6A	001764	991	994#											
LK6.1	002060	985	1007#											
LK7	002106	1011	1016#	1021										
LK8	002162	1028	1030#											
LK8A	002224	1043#												
LK8B	002240	1065#												
LK8C	002300	1075#	1080											
LK8D	002314	1076	1079#											
LK8E	002356	1090#	1095											
LK9	002404	1091	1094#											
LK9B	002500	1099	1102	1105	1108#									
LK9C	002512	1121#												
LK9E	002606	1123	1137#											
LK9F	002620	1136	1140#											
LK9G	002724	1142	1156#											
LOCTRP	001312	1155	1159#											
LODAD	007442	1172	1175	1178	1181#									
LODIV	007406	905#	1842*	1856*	1857									
LODMSC	007464	2047	2059	2061#										
LOPSW =	040000	2049	2053#											
MINCLK =	000010	2051	2066#											
		2451#	2465											
		855#	1235	2206	2208	2212	2215	2219						







TIMOUT	001270	896#	1547	1588	1769	1975*	2066*	2147										
TIMRUN	001262	893#	1134	1153														
TKB	000406	789#	2478	2548	2549													
TKS	000404	788#	2476	2546														
THINCLK	003124	1235#	1241															
TOSN =	000040	848#	1354	1357	1358	1445	1448	1449	1574	1583								
TPB	000412	791#	2549*	2574*														
TPS	000410	790#	2556	2572														
TRANST	001314	906#	1786															
TRIGGER	001272	897#	1539	1542	1580	1644	1763	1980*	2067*	2157*								
TSTABL	012662	914	2038	2040	2506#													
TST1	001334	920#	925	2505														
TST10	003070	1225#	1230															
TST11	003162	1253#	1258															
TST12	003242	1275#	1280															
TST13	003404	1313#	1318															
TST14	003636	1366#	1371															
TST15	004022	1408#	1413															
TST16	004274	1467#	1472															
TST17	004452	1509#	1514															
TST2	001530	967#	972															
TST20	004674	1563#	1568															
TST21	005014	1594#	1599															
TST22	005332	1678#	1683															
TST23	005552	1734#	1739															
TST24	006046	1806#	1811															
TST25	006102	1819#	1824															
TST26	006312	1873#	1878															
TST3	001672	1000#	1005															
TST4	002062	1036#	1041															
TST5	002136	1056#	1061															
TST6	002360	1114#	1119															
TST7	002726	1187#	1192															
TTRIG	001300	900#	2147*	2149	2153*	2154												
TTY = %000005		766#	2457	2458*	2460*	2560	2563*	2564	2566	2568	2570	2574	2576*	2577				
TTYFLG	013140	2578#	2580*	2583*	2607*	2609*	2611*											
TYPE = 000004		2550	2555#	2557	2584													
		771#	1217	1267	1812	1879	1894	1896	1902	1915	1928	1937	1947	1955				
		1971	1976	1981	2017	2025	2193	2287	2453	2536	2625	2676						
T.ACCE	012762	802	2510#															
T.ERRO	012434	799	2452#															
T.KEY.	013106	803	2545#															
T.RSTR	013510	801	2643#															
T.SAVR	013446	800	2629#															
T.1	007532	2078#	2079	2082														
VOID	001206	866#	2062*	2258*														
WDADOR	010576	1945	2256#															
WDBASE	001306	903#	1944*	2058	2256	2257*	2258	2259*	2260	2261*	2262							
WDCSR	001210	867#	1286#	1287	1324*	1354	1357*	1358	1377*	1378	1381*	1382	1386*	1387				
		1390#	1391	1395*	1396	1399*	1400	1419*	1425	1428*	1442	1445	1448*	1449				
		1456#	1477*	1519*	1520	1523*	1528	1531*	1534	1543*	1548	1573*	1574	1583				
		1614#	1615	1618	1621*	1622	1645	1652	1654	1661	1665	1668*	1696*	1700				
		1703#	1704*	1715	1718	1724*	1752*	1753	1756*	1764	1767*	1772*	1773	1776*				
		1792	1796*	1837*	1838	1841*	1849*	2063*	2260*									
WDDAT	001212	868#	1235	1293*	1294	1297	1300*	1301	1304	1325*	1326	1329	1332*	1333				
		1336	1340*	1341	1344	1347*	1348	1351	1429*	1430*	1431*	1433*	1478*	1479				









ADD	1100	1173	2039	2187	2188	2205	2249	2257	2259	2261	2268	2270	2272	2442	2582
ASL	1959	1960	1961	1962	1963	2037	2183	2185	2186	2233	2234	2247	2248	2530	2531
ASR	2532														
BEQ	2250	2251													
	828	835	928	958	1022	1026	1099	1102	1105	1172	1175	1178	1291	1302	1334
	1349	1359	1383	1392	1401	1426	1450	1480	1494	1521	1535	1549	1575	1584	1616
	1619	1630	1638	1648	1655	1666	1701	1754	1774	1793	1839	1858	1912	1934	1943
	1950	1958	1974	1979	1984	2010	2020	2096	2110	2124	2130	2139	2155	2173	2175
	2177	2211	2213	2216	2218	2238	2462	2466	2494	2520	2522	2524	2565	2567	2569
BGE	2571	2614	2617												
BGT	2091	2105	2113	2134											
BIC	2181	2528													
	1135	1154	1209	1300	1332	1347	1357	1381	1390	1399	1430	1433	1448	1456	1525
	1531	1545	1578	1582	1628	1636	1668	1706	1724	1758	1767	1796	1846	1849	1966
	2080	2182	2276	2286	2292	2441	2480	2529	2551						
BIS	1293	1325	1340	1377	1386	1395	1428	1429	1431	1523	1524	1543	1544	1577	1581
	1621	1632	1703	1704	1705	1756	1757	1776	1841	1845	2275	2282	2533		
BISB	2618														
BIT	834	927	948	957	1235	1294	1297	1301	1304	1326	1329	1333	1336	1341	1344
	1348	1351	1354	1358	1378	1382	1387	1391	1396	1400	1425	1445	1449	1479	1482
	1486	1489	1493	1496	1520	1528	1534	1548	1574	1583	1615	1622	1633	1637	1652
	1654	1661	1665	1700	1718	1753	1764	1773	1792	1838	2078	2206	2208	2212	2215
	2219	2461	2465	2484	2486	2561	2600								
BLOS	2438														
BLT	2179	2526													
BMI	932	938	989	1049	1443	1716	2232								
BNE	832	940	949	983	1080	1095	1166	1169	1216	1219	1236	1241	1243	1246	1269
	1295	1298	1305	1327	1330	1337	1342	1345	1352	1355	1379	1388	1397	1437	1446
	1461	1483	1487	1490	1497	1503	1529	1555	1623	1634	1653	1662	1672	1710	1719
	1728	1765	1800	1853	1867	1905	1909	1914	1986	1992	1998	2004	2028	2034	2082
	2085	2088	2102	2116	2150	2170	2207	2209	2220	2226	2229	2245	2285	2482	2485
	2487	2517	2562	2601	2620	2622	2675								
BPL	945	954	961	979	1010	1045	1070	1073	1076	1085	1088	1091	1127	1130	1146
	1149	1214	1263	1420	2079	2097	2111	2125	2140	2156	2190	2239	2477	2547	2557
	2573														
BR	985	1136	1155	1238	1439	1712	1855	1907	1918	1990	1996	2002	2008	2015	2086
	2100	2114	2128	2143	2192	2194	2214	2221	2283	2534	2537	2575	2579	2581	2597
	2605														
CLR	935	936	943	952	956	977	981	1008	1043	1046	1065	1066	1068	1071	1074
	1078	1081	1082	1083	1086	1089	1093	1122	1125	1128	1141	1144	1147	1159	1212
	1232	1233	1261	1286	1324	1419	1435	1457	1477	1478	1500	1519	1552	1573	1614
	1625	1669	1696	1708	1725	1752	1772	1797	1837	1842	1850	1864	2024	2089	2103
	2117	2132	2148	2163	2166	2227	2500	2511	2514	2598	2672				
CLRB	2604	2606	2624												
CMP	1021	1025	1098	1101	1104	1165	1168	1171	1174	1177	1857	1985	1991	1997	2003
	2009	2084	2087	2090	2101	2104	2115	2118	2129	2133	2149	2189	2225	2228	2231
	2244	2493	2621												
CMPB	1290	1618	1629	1647	1904	1908	1911	1913	2027	2169	2172	2174	2176	2178	2180
	2437	2481	2516	2519	2521	2523	2525	2527	2564	2568	2570				
DEC	831	1460	1502	1554	1671	1727	1799	1852	1866	1965	2081	2210	2217	2674	
EMT	799	800	801	802	803										
HALT	608	609	611	613	615	617	619	621	623	625	627	629	631	633	635
	637	639	641	643	645	647	649	651	653	655	657	659	661	663	665
	667	669	671	673	675	677	679	681	683	685	687	689	691	693	695
	697	699	701	703	705	707	709	711	713	715	717	719	721	723	725
	727	729	731	733	735	737	2463	2666							



.MAIN. MACY11 27(732) 15-OCT-76 11:28 PAGE 79  
 DBQAAA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.EVEN	1924	2428	2589	2683											
.LIST	1	562	597	796	799	800	801	802	803	914	917	925	964	972	997
	1005	1033	1041	1053	1061	1111	1119	1184	1192	1222	1230	1250	1258	1272	1280
	1310	1318	1363	1371	1405	1413	1464	1472	1506	1514	1560	1568	1591	1599	1675
.MACRO	1623	1731	1739	1803	1811	1816	1824	1870	1878						
	1	563	564	565	566	568	571	573	575	578	579	591	583	584	585
.NLIST	587	588	589	593	594	595	597								
	1	562	597	796	799	800	801	802	803	914	917	925	964	972	997
	1005	1033	1041	1053	1061	1111	1119	1184	1192	1222	1230	1250	1258	1272	1280
	1310	1318	1363	1371	1405	1413	1464	1472	1506	1514	1560	1568	1591	1599	1675
.PAGE	1683	1731	1739	1803	1811	1816	1824	1870	1878						
	917	964	997	1033	1053	1111	1184	1222	1250	1272	1310	1363	1405	1464	1506
	1560	1591	1675	1731	1803	1816	1870								
.REM	1	599													
.REPT	610														
.SBTTL	917	964	997	1033	1053	1111	1184	1222	1250	1272	1310	1363	1405	1464	1506
	1560	1591	1675	1731	1803	1816	1870	1883	2295	2429					
.WORD	2592														

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DBQAAA.SEQ/SOL/CRF/NL:TOC/PAGNUM/DS:ERFZ=SYSMAC.CO,DBQAAA.P11  
 RUN-TIME: 32 41 4 SECONDS  
 RUN-TIME RATIO: 187/78=2.3  
 CORE USED: 41K (81 PAGES)

