KW11K

DIAGNOSTIC

MD-11-DZKWK-A

EP-DZKWK-A-DL-A

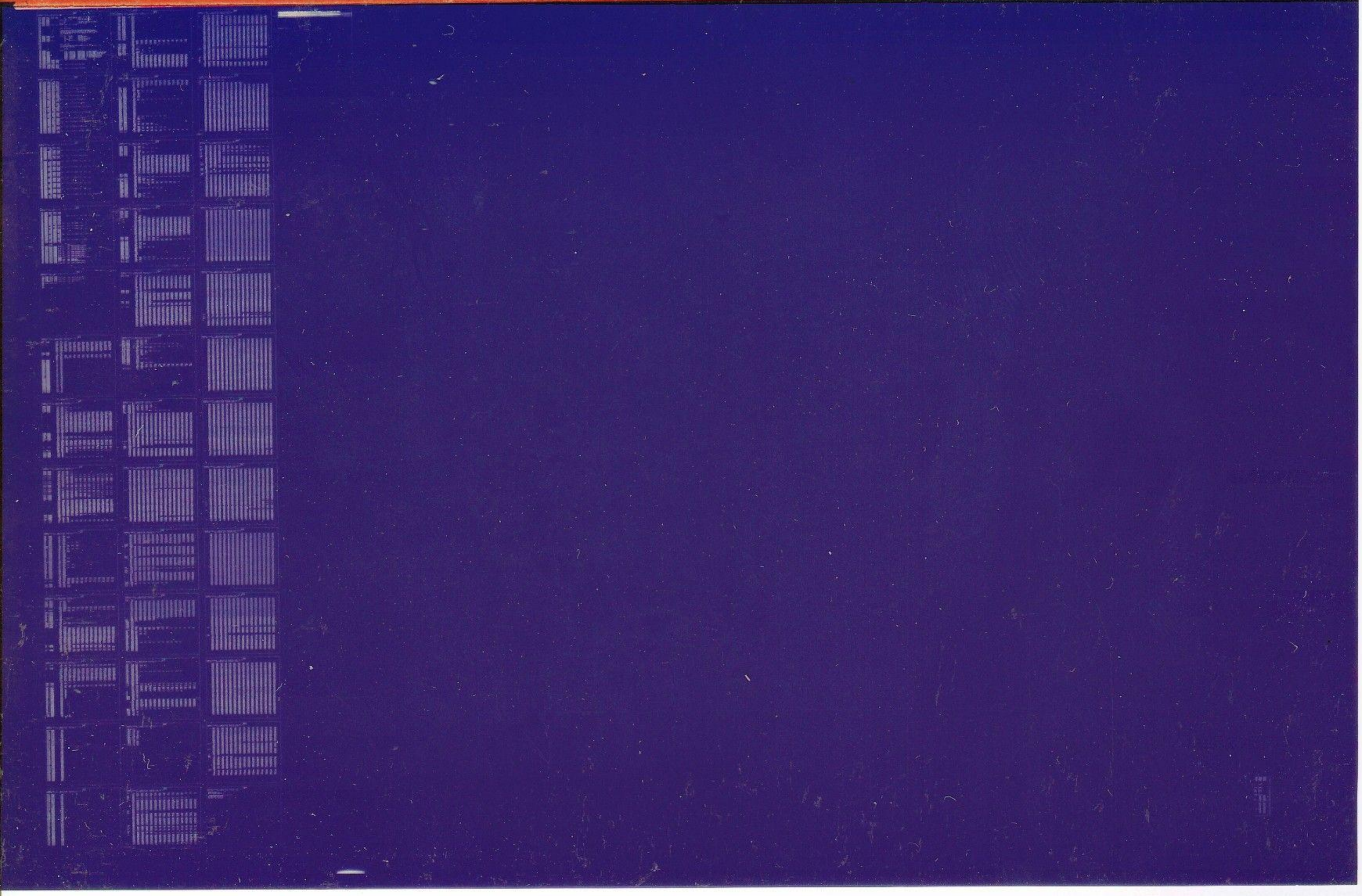
COPYRIGHT © 1976

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA



KW11-K

DIAGNOSTIC

MD-11-DZKWK-A

EP-DZKWK-A-DL-A

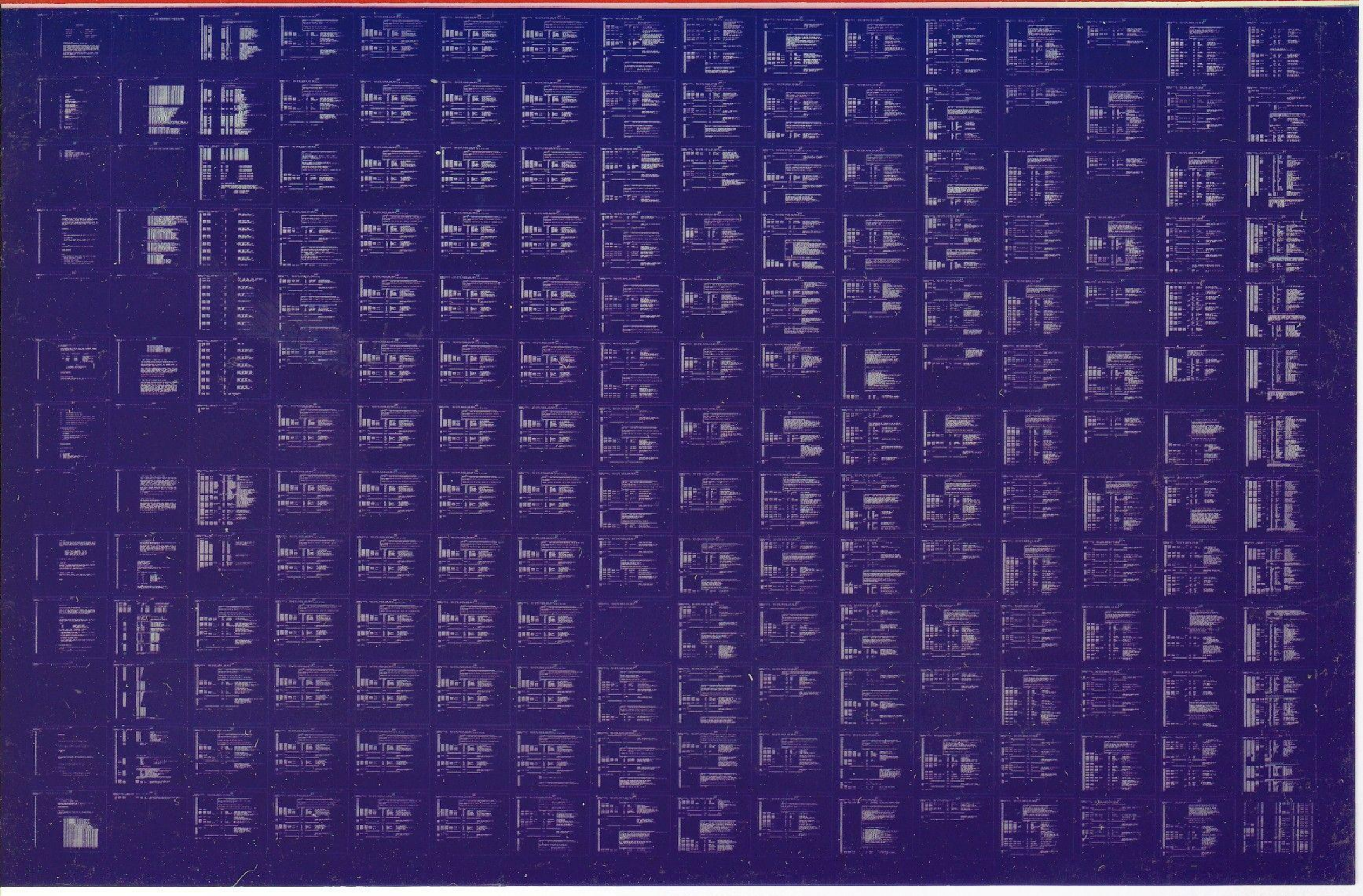
COPYRIGHT © 1976

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA



```
.REM    %

.REM    +
```

IDENTIFICATION
****

PRODUCT CODE:	MAINDEC-11-DZKWK-A-D
PRODUCT NAME:	KW11K DIAGNOSTIC
DATE CREATED:	MARCH 1, 1976
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	EDWARD C. BADGER

COPYRIGHT (C) 1976
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

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

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

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

TABLE OF CONTENTS
****

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.0	LOADING PROCEDURE
3.1	METHOD
3.2	NON-STANDARD ADDRESS, VECTOR, OR PRIORITY; OR USE OF SOFTWARE SWR
4.0	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESSES
4.3	PROGRAM AND/OR OPERATOR ACTION
5.0	OPERATING PROCEDURE
5.1	SWITCH REGISTER FUNCTION
5.2	SCOPE LOOPS
5.3	PROGRAM AND/OR OPERATOR ACTION
5.3.1	LOGIC TEST
5.3.2	SPECIAL I/O SIGNAL TESTS
6.0	ERRORS
6.1	ERROR PRINTOUT
6.1.1	EXAMPLE
6.2	NON-STANDARD ERROR HALTS
7.0	RESTRICTIONS
7.1	
7.2	
8.0	MISCELLANEOUS
8.1	
8.2	
8.3	EXECUTION TIME
8.3.1	LOGIC TEST
8.3.2	SPECIAL I/O SIGNAL TESTS

9.0	PROGRAM DESCRIPTION	
9.1	LOGIC TESTS	
9.2	SPECIAL EXTERNAL I/O SIGNAL TESTS	
9.2.1	LS210 "STP2 OUT" TO "SCHMITT TRIG 1 IN" TESTS	
9.2.2	LS214 "STP1 OUT" TO "SCHMITT TRIG 2" H TESTS	
9.2.3	LS220 "SCHMITT TRIG 3" IN, "ST3 OUT" TESTS	
9.2.4	LS224 "A EVENT OUT" TEST	
9.2.5	LS230 "B EVENT OUT" TEST	
10.0	LISTING TABLE OF CONTENTS	
11.0	LISTING	

1.0 ABSTRACT

THIS PROGRAM ALLOWS THE USER TO CHECK OUT OR DEBUG THE KW11K DUAL REAL TIME CLOCK. THE LOGIC TEST IS SELF CONTAINED AND NEEDS NO EXTERNAL MAINTENANCE HARDWARE OR OPERATOR INTERVENTION.

FIVE SPECIAL TESTS ARE INCLUDED WITHIN THIS PROGRAM TO ALLOW THE USER TO CHECK OUT AND DEBUG THE EXTERNAL I/O SIGNALS. TO RUN THESE TESTS A JUMPER WIRES IS NEEDED IN ORDER TO LOOP OUTPUT TO AN INPUT.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP11 FAMILY COMPUTER WITH 8K OF MEMORY OR MORE AND I/O FACILITIES (A SWITCH REGISTER OR TTY).
2. KW11K UNDER TEST.
3. FOR EXTERNAL I/O SIGNAL TESTS A LOOPBACK WIRE (JUMPER) IS NEEDED. JUMPERS ARE 30 AWG JUMPER TYPE 915.

2.2 STORAGE

THIS PROGRAM OCCUPIES AND USES ONLY THE LOWER 8K OF MEMORY.

3.0 LOADING PROCEDURE

3.1 METHOD

STANDARDS PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

1. ABSOLUTE LOADER MUST BE IN MEMORY.
2. PLACE BINARY TAPE IN READER.
3. LOAD ADDRESS *7500 (* DETERMINED BY LOCATION OF LOADER).
4. PRESS "START" (PROGRAM WILL BE LOADED INTO MEMORY).

```
  167
  168                    THE PROGRAM CAN ALSO BE LOADED BY XXDP, ACT, OR APT.
```

3.2 NON-STANDARD ADDRESS, VECTOR, OR PRIORITY; OR USE OF SOFTWARE SWITCH REGISTER

THIS PROGRAM IS SET TO TEST A KW11K WITH A STANDARD ADDRESS, VECTOR, AND PRIORITY. IF ANY OF THESE ARE DIFFERENT ON THE KW11K YOU ARE TESTING, CHANGE THE CORRESPONDING LOCATION IN MEMORY BEFORE STARTING THIS TEST.

LOCATION	TAG	CURRENT CONTENTS	COMMENTS
1254	$BASE:	170404	;;BASE ADDRESS OF EQUIPMENT ;; UNDER TEST
1250	$VECT1:	000344	;;INTERRUPT VECTOR #1
1252	$PRIOR:	000006	;;BUS PRIORITY - 1,#2
	SWREG:	000000	;MANUAL SWR.
	$TPFLG:	.BYTE 0	;;"TERMINAL AVAILABLE" ; FLAG (BIT<0:7>=0=YES)

NOTE

IF NO HARDWARE SWITCH REGISTER EXISTS,
YOU MAY SET ANY BIT IN "SWREG" AS YOU
WOULD HAVE SET IT IN THE SWR.

4.0 STARTING PROCEDURE
****

4.1 CONTROL SWITCH SETTINGS

STARTING AT MEMORY LOCATIONS 200, 204, 210, 214, 220, 224, OR 230
SET ALL SWITCHES AS DESIRED. SEE SECTION 5.1.

4.2 STARTING ADDRESSES

200 START ADDRESS FOR LOGIC TEST.

204 RESTART ADDRESS FOR LOGIC TEST.

210 START ADDRESS FOR "STP2 OUT", "SCHMITT TRIG 1" TESTS.

214 START ADDRESS FOR "STP1 OUT", "SCHMITT TRIG 2" TESTS.

220 START ADDRESS FOR "SCHMITT TRIG 3 IN", "ST3 OUT" TESTS.

224 STARTING ADDRESS FOR "A EVENT OUT" TEST.

230 STARTING ADDRESS FOR "B EVENT OUT" TEST.

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO CORE.
2. SET SWITCH REGISTER TO STARTING ADDRESS.
3. LOAD ADDRESS.
4. SET SWITCHES TO DEISRED SETTINGS - SEE SECTION 5.1.
5. IF STARTING A SPECIAL I/O SIGNAL TEST:
   MAKE WIRE LOOP CONNECTION.
6. PRESS START.

5.0 OPERATING PROCEDURE
****

5.1 SWITCH REGISTER FUNCTION

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUT (ALL TESTS)
13	INHIBIT "*" TYPEOUT (SPECIAL I/O SIGNAL TESTS)
11	INHIBIT ITERATIONS (SHORT PASS)
10	BELL ON ERROR

```
                                9     LOOP ON ERROR
                                8     LOOP ON TEST IN SWR <7:0>
```

5.2 SCOPE LOOPS

IF AN ERROR OCCURS AND THE USER WISHES TO SCOPE THE ERROR, HE (OR SHE) SHOULD SET SW15=1 TO HALT ON ERROR. THEN WHEN THE PROGRAM HALTS ON ERROR, SW15=0, SET SW14=1. TO LOOP ON CURRENT TEST, SET SW13=1 TO INHIBIT ERROR PRINTOUT, AND PRESS CONTINUE ON THE CPU'S CONSOLE.

NOTE

FOR EACH TEST IN THE LISTING, YOU WILL FIND A TEST DESCRIPTION. IN EACH DESCRIPTION A PROBABLE SYNC POINT IS LISTED. THESE POINTS ARE LISTED AS A GUIDE IN ORDER FOR YOU TO SYNC YOUR SCOPE TO THE SIGNALS BEING GENERATED.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOGIC TEST

THE FIRST PASS THROUGH THE PROGRAM WILL BE MADE WITH ITERATIONS INHIBITED. SUCCESSIVE PASSES WILL ENABLE ITERATIONS IF SWR11=0. "END PASS" IS PRINTED OUT AT THE END OF A PASS.

IF NOT INHIBITED BY APT, THE PROGRAM WILL LOOK FOR MORE KW11KS TO EXERCISE, ONE PASS WILL EXERCISE ALL KW11KS.

5.3.2 SPECIAL I/O SIGNAL TESTS

THERE ARE NO "SHORT PASSES". EACH PASS WILL ITERATE 65,324 TIMES. A "*" IS TYPED AT THE END OF A PASS UNLESS SWR13=1.

6.0 ERRORS
******

6.1 ERROR PRINTOUT

PRINTOUT VARIES WITH THE ERROR DETECTED. THE ERROR PC TYPED OUT IS HE ACTUAL LOCATION OF THE ERROR CALL.

A HALT AT LOCATION "$TYPE"+10 WHEN RUNNING WITH NO TERMINAL INDICATES AN ERROR HAS OCCURRED. TO FIND OUT THE NUMBER OF THE ERROR, EXAMINE LOCATION "$TSTNM". THIS IS THE ITEM NUMBER OF THE ERROR. TO FIND OUT WHAT THE ERROR TYPOUT WOULD HAVE BEEN GOTO TO THE ERROR POINTER TABLE BEGINNING AT LOCATION "$ERRTB".

6.1.1 EXAMPLE

IF WE EXAMINED LOCATION "$TSTNM" AND FOUND A 5 (101) WE GO TO LOCATION "$ERRTB" AND LOOK THROUGH THE ERROR POINTER TABLE UNTIL WE FOUND ITEM 5. THE INFORMATION WOULD LOOK LIKE:

```
;ITEM 5

     EMS                ;CLOCK B SR DATA ERROR
     DH5                ;ERRPC   BSR     WAS     S/B
     DT5                ;$ERRPC,BSR,$BDDAT,$GDDAT
     DF0                ;ALL NUMBERS ARE IN OCTAL FORM
```

TO FIND OUT THE INFORMATION SPECIFIED BY DT5 (SERRPC,BSR,$GDADR,$BDADR) FOLLOW THESE STEPS:

1. LOOK UPHE ADDRESS OF THE LABEL (I.E., $ERRPC) IN THE SYMBOL TABLE WHICH FOLLOWS THE LISTING.
2. PUT THIS ADDRESS IN THE WITCH REGISTER AND DEPRESS THE LOAD ADDRESS SWITCH ON THE PROCESSOR'S CONSOLE.
3. NOW DEPRESS THE EXAMINE SWITCH.
4. THE DATA DISPLAYED IN THE DATA LIGHTS IS THE INFORMATION THAT WOULD HAVE BEEN PRINTED FOR HIS LABEL IF YOU HAD A INPUT/OUTPUT TERMINAL.

6.2 NON-STANDARD ERROR HALTS

ANY HALT IN THE TRAP CATCHER AREA LOCATIONS 000000-001000.

INDICATES:

1. THE KW11K INTERRUPTED TO A WRONG VECTOR ADDRESS.

   OR

371 2. TIME-OUT OR ILLEGAL INSTRUCTION HARDWARE TRAP.

7.0 RESTRICTIONS
****

7.1

JUMPER W2 MUST BE INSTALLED IF NOT JUMPERED ON MODULE.

7.2

LOGIC TEST MUST BE RUN BEFORE ANY SPECIAL I/O SIGNAL TEST.

8.0 MISCELLANEOUS
****

8.1

AFTER A POWER FAILURE OCCURS, PROGRAM EXECUTION WILL CONTINUE AT THE POINT WHERE THE POWER FAILURE OCCURED AFTER THE PROGRAM TYPES "POWER".

8.2

THIS PROGRAM IS CHAINABLE UNDER XXDP, ACT, OR APT.

8.3 EXECUTION TIME

8.3.1 LOGIC TEST

.3 MINUTES (20 SEC.) ITERATIONS INHIBITED - NO ERRORS.

4.0 MINUTES (240 SEC.) WITH ITERATIONS - NO ERRORS.

8.3.2 SPECIAL I/O SIGNAL TESTS

1.0 MINUTES NO ERRORS, SW13=0.

EXECUTION TIMES ARE APPROXIMATE, AS THE VARIOUS PDP-11 CPU'S HAVE VARIED INSTRUCTION EXECUTION TIMES.
TIMES QUOTED WERE TAKEN FROM A RUN ON A PDP-11/05.

9.0 PROGRAM DESCRIPTION
****

9.1 LOGIC TESTS

A COMPLETE DESCRIPTION OF EACH TEST IS INCLUDED WITHING THE LISTING BEFORE EACH TEST. BELOW IS A LIST OF TESTS PREFORMED ON THE KW11K.

*
* PHASE 1 CLOCKS A+B BASIC LOGIC TESTS.
*

*TEST THE ADDRESSABILITY OF CLOCK A
*TEST THE ADDRESSABILITY OF CLOCK A'S BUFFER REG.
*TEST THE ADDRESSABILITY OF CLOCK A'S COUNT REG.
*TEST THE ADDRESSABILITY OF CLOCK B'S CSR
*TEST THE ADDRESSABILITY OF CLOCK B'S BUFFER REG.
*TEST THE ADDRESSABILITY OF CLOCK B'S COUNT REG.
*TEST THAT CLOCK A BUFFER CAN BE WRITTEN INTO
*TEST THAT CLOCK A BUFFER CAN BE WRITTEN TO A ZERO
*TEST THAT CLOCK A'S STATUS CAN BE WRITTEN AND READ
*TEST THAT CLOCK B'S STATUS REGISTER CAN BE WROTE/READ
*TEST THAT CLOCK B'S BUFFER REGISTER CAN BE WROTE/READ
*TEST THAT CLOCK A STATUS REGISTER BIT 15 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 14 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 13 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 9 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 8 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 7 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 6 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 5 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 3 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 2 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 1 CAN BE SET AND CLEARED
*TEST THAT CLOCK A STATUS REGISTER BIT 0 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 0 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 1 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 2 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 3 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 4 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 5 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 6 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 7 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 8 CAN BE SET AND CLEARED

```
*TEST THAT CLOCK A BUFFER REGISTER BIT 9 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 10 CAN BE SET AND CLEARED
*TEST THAT CLOCK A BUFFER REGISTER BIT 11 CAN BE SET AND CLEARED
```

```
                        *TEST THAT CLOCK A BUFFER REGISTER BIT 12 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK A BUFFER REGISTER BIT 13 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK A BUFFER REGISTER BIT 14 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK A BUFFER REGISTER BIT 15 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 11 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 7 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 6 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 5 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 4 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 3 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 2 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 1 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B STATUS REGISTER BIT 0 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 0 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 1 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 2 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 3 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 4 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 5 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 6 CAN BE SET AND CLEARED
                        *TEST THAT CLOCK B BUFFER REGISTER BIT 7 CAN BE SET AND CLEARED
                *
                * PHASE 2 ADVANCED BASIC LOGIC TESTS
                *
                        *TEST THE LOW BYTE OPERATION OF CLOCK A'S STATUS REGISTER
                        *TEST THE HIGH BYTE OPERATION OF A'S STATUS REGISTER
                        *TEST THE LOW BYTE OPERATION OF B'S STATUS REGISTER
                        *TEST THE HIGH BYTE OPERATION OF B'S STATUS REGISTER
                        *TEST THAT CLOCK A'S COUNT REGISTER IS CLEAR
                        *TEST CLOCK A'S COUNT REGISTER WITH 125252 PATTERN
                        *TEST CLOCK A'S COUNT REGISTER WITH 052525 PATTERN
                        *TEST THAT CLOCK B'S COUNT REGISTER IS CLEAR
                        *TEST CLOCK B'S COUNT REGISTER WITH 125 PATTERN
                        *TEST CLOCK B'S COUNT REGISTER WITH 252 PATTERN
                        *TEST THAT INIT CLEARS STATUS REGISTER A
                        *TEST THAT INIT CLEARS BUFFER REGISTER A
                        *TEST THAT INIT CLEARS STATUS REGISTER B
                        *TEST THAT INIT CLEARS BUFFER REGISTER B
                        *TEST THE SETTING OF MAINTENANCE STP1 IN CLOCK A BIT 15 TO SET
                        *TEST THAT BIT00 IN CLOCK A STATUS REG. WILL SET WHEN BI  3 AND MAIN. STP
                        *TEST THAT CLOCK A WILL INCREMENT - MODE 0 - RATE STP1 FIRST COUNT TEST
                        *TEST THE ABILITY OF CLOCK A TO COUNT FROM ZERO TO OVERFLOW USING M STP1'
                *
                * PHASE 3 CLOCK A COUNT FUNCTION TESTS
                *
                        *TEST THAT CLOCK A OVERFLOW WILL OCCUR
                        *TEST IN CLOCK A THAT OVERFLOW IN MODE 0 CAUSE CLEARING OF "ENB CNTR" F/F
                        *TEST THE ABILITY OF CLOCK A TO COUNT AT 1MHZ RATE PART 1
                        *TEST THE ABILITY OF CLOCK A TO COUNT AT 100KHZ RATE PART 1
                        *TEST THE ABILITY OF CLOCK A TO COUNT AT 10KHZ RATE PART 1
                        *TEST THE ABILITY OF CLOCK A TO COUNT AT 1KHZ RATE PART 1
                        *TEST THE ABILITY OF CLOCK A TO COUNT A 100HZ RATE PART 1
                        *TEST THE ABILITY OF CLOCK A TO COUNT AT LINE-FREQ RATE PART 1
```

```
535                                     *TEST THAT CLOCK A DOESN'T COUNT WHEN NO RATE IS SELECTED
536                                     *TEST THAT CLOCK A'S COUNT REG ISN'T LOADED WHEN CLOCK A IS ENABLED
```

```
                        *TEST THAT CLOCK A IN MODE 1 DOES NOT CLEAR ENABLE ON OVERFLOW
                        *TEST THAT A CLOCK A "BUFFER TO COUNT REG" DOESN'T TAKE PLACE ON A MODE 2
                        *TEST THAT CLOCK A MODE 2 + MAINTENANCE S12 SET MODE FLG
                        *TEST THAT PATTERN 052525 CAN BE XFERRED BETWEEN A'S COUNT-BUFFER REGS
                        *TEST THAT PATTERN 125252 CAN BE XFERRED BETWEEN A'S COUNT-BUFFER REGS
                        *TEST THAT A'S COUNT REG. IS CLEARED BY INIT
                        *TEST THAT A'S COUNT REGISTER ISN'T CLEARED IN MODE 1 WHEN STP2 IS GENERA
                        *TEST THAT A'S COUNT REGISTER ISN'T CLEARED IN MODE 2 WHEN STP2 IS GENERA
                        *TEST THAT MODE 3+ "STP2" CLEARS A'S COUNT REGISTER
                        *TEST THE AUTODECREMENT FEATURE OF CLOCK A'S BUFFER
                        *TEST THAT CLOCK A'S 1MHZ CLR CAN BE DISABLED
                *
                * PHASE 4 CLOCK B COUNT FUNCTION TESTS
                *
                        *TEST THAT CLOCK B WILL COUNT ONCE FIRST CLOCK B COUNT
                        *TEST THE ABILITY IF CLOCK B TO COUNT FROM ZERO TO OVERFLOW
                        *TEST THAT CLOCK B CAN GENERATE AN OVERFLOW
                        *TEST THE INIT. ABILITY OF CLOCK B'S COUNT REG.
                        *TEST THAT CLOCK B DOESN'T COUNT WHEN NO RATE IS SELECTED
                        *TEST THE ABILITY OF CLOCK B TO COUNT AT 1MHZ PART 1
                        *TEST THE ABILITY OF CLOCK B TO COUNT AT 100KHZ PART 1
                        *TEST THE ABILITY OF CLOCK B TO COUNT AT 10KHZ PART 1
                        *TEST THE ABILITY OF CLOCK B TO COUNT AT 1KHZ PART 1
                        *TEST THE ABILITY OF CLOCK B TO COUNT AT 100HZ PART 1
                        *TEST THE ABILITY OF CLOCK B TO COUNT AT LINE-FREQ PART 1
                        *TEST THE "FEED B TO A" 24 BIT COUNTER FEATURE OF CLOCKS A + B
                *
                * PHASE 5 CLOCKS A+B INTERRUPT TESTS
                *
                        *TEST THAT CLOCK A WILL INTR. AND TO RIGHT VECTOR
                        *TEST THAT CLOCK A WILL INTR WHEN CPU PSW = CLK INTR LEV -1
                        *TEST THAT CLOCK A WILL NOT INTR. WHEN CPU PSW=CLK INTR LEVEL
                        *TEST THAT S   WILL CAUSE CLOCK A TO INTR.
                        *TEST THAT CLOCK I OVERFLOW WILL CAUSE AN INTERRUPT
                        *TEST THAT A CLOCK A COUNTER BUFFER CAUSES AN INTERRUPT
                        *TEST THAT CLOCK B WILL INTR. AND TO RIGHT VECTOR
                        *TEST THAT CLOCK A WILL INTR WHEN CPU PSW CLK INTR LEV -1
                        *TEST THAT CLOCK B WILL NOT INTR. WHEN CPU PSW=CLK INTR LEVEL
                        *TEST THAT A CLOCK B OVERFLOW CAUSES ON INTERRUPT
                *
                * PHASE 6 CLOCK A+B ADVANCE TESTING
                *
                        *TEST THAT THE TRAILING EDGE OF STP1 WILL INCR. COUNTER
                        *TEST CLOCK A'S 100KHZ DIVIDER
                        *TEST CLOCK A'S 10KHZ DIVIDER
                        *TEST CLOCK A'S 1KHZ DIVIDER
                        *TEST CLOCK A'S 100HZ DIVIDER
                        *TEST CLOCK A'S REPEATIBILITY AT 1MHZ RATE
                        *TEST CLOCK A'S REPEATIBILITY AT 100KHZ RATE
                        *TEST CLOCK A'S REPEATIBILITY AT 10KHZ RATE
                        *TEST CLOCK A'S REPEATIBILITY AT 1KHZ RATE
                        *TEST CLOCK A'S REPEATIBILITY AT 100HZ RATE
                        *TEST CLOCK B'S 100KHZ DIVIDER
```

```
593                                     *TEST CLOCK B'S 10KHZ DIVIDER
```

```
                        *TEST CLOCK B'S 1KHZ DIVIDER
                        *TEST CLOCK B'S 100HZ DIVIDER
                        *TEST THAT "INIT" CLEARS CLOCK B'S 100HKZ DIVIDE BY 10 CHIPS
                        *TEST CLOCK B'S REPEATIBILITY AT 1MHZ RATE
                        *TEST CLOCK B'S REPEATIBILITY AT 100HKZ RATE
                        *TEST CLOCK B'S REPEATIBILITY AT 10KHZ RATE
                        *TEST CLOCK B'S REPEATIBILITY AT 1KHZ RATE
                        *TEST CLOCK B'S REPEATIBILITY AT 100HZ RATE


9.2  SPECIAL EXTERNAL I/O SIGNAL TESTS


9.2.1  LS210 "STP2 OUT" TO "SCHMITT RIG 1 IN" TESTS

     THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND PROVIDING SCOPE
     LOOP CAPABILITIES FOR "STP2 OUT" L AND "SCHMITT TRIG 1" IN.

     WHEN YOU LOAD AND START AT LOCATION 210, PROGRAM CONTROL IS
     TRANSFERRED HERE.  "STP2 OUT" L PULSES ARE GENERATED BY "LO STAT
     A HI" H + "BD10" H (MAIN.  STP2).

     PIN V ("STP2 OUT") IS WIRED TO PIN LL (SCHMITT TRIG1) FOR THIS
     TEST.  "STP2 OUT" PULSES ARE RECEIVED AS "SCHMITT TRIG 1" PULSES
     WHICH SET CLOCK A'S STATUS REGISTER BIT 15.  IF AN ERROR IS
     DETECTED, NORMAL ERROR REPORTING TECHNIQUE.  AND ERROR SWITCH
     REGISTER OPTIONS ARE USED.  AN "*" IS TYPED AFTER EACH 65,324
     LOOPS THROUGH THE TEST.  SW13=1 WILL INHIBIT THIS FEATURE.

     YOU MUST WIRE PINS V AND LL OF J1 TOGETHER.

     LOGIC TEST (L + S 200) SHOULD BE RUN FIRST.


9.2.2  LS214 "STP1 OUT" TO "SCHMITT RIG 2" H TESTS

     THIS IS A SPECIAL TEST SECTION DEVOTED FOR TESTING AND PROVIDING
     SCOPE LOOP CAPABILITIES FOR "STP2 OUT" AND "SCHMITT TRIG2" IN.

     WHEN YOU LOAD AND START AT LOCATION 214, PROGRAM CONTROL IS
     TRANSFERRED HERE.  "STP1 OUT" L PULSES ARE GENERATED BY "LD STAT
     A HI" + "BD12" H (MIN S ).  PIN DD ("STP1 OUT") IS WIRED TO PIN
     BB ("SCHMITT TRIG 2") FOR THIS TEST.  "STP1 OUT" PULSES ARE
     RECEIVED AS "SCHMITT RIG 2" PULSES WHICH WILL CLEAR CLOCK A'S
     COUNT REGISTER IF MODE 3 IS SELECTED.  IF AN ERROR IS DETECTED,
     NORMAL ERROR REPORTING TECHNIQUE.  AND ERROR SWITCH REGISTER
     OPTIONS ARE USED.  AN "*" IS TYPED AFTER EACH 65,324 LOOPS
     THROUGH THE TEST.  SW13=1 WILL INHIBIT THIS FEATURE.

     YOU MUST WIRE PINS DD AND BB OF J1 TOGETHER.
```

```
650
651                     LOGIC TESTS (L + S AT 200) SHOULD BE RUN FIRST.
```

9.2.3 LS220 "SCHMITT TRIG 3" IN, "ST3 OUT" TESTS

THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND PROVIDING SCOPE LOOPS CAPABILITIES FOR "SCHMITT TRIG 3" AND "ST3 OUT".

WHEN YOU LOAD AND START AT LOCATION 220, PROGRAM CONTROL IS TRANSFERRED HERE. "STP2" PULSES ARE GENERATED BY "LD STAT A H," + "BD10" H (MAIN STP2). PIN V ("STP2 OUT") IS WIRED TO PIN T ("SCHMITT RIG 3"). "SCHMITT TRIG 3" PULSES GIVE US "ST3 OUT" PULSES. PIN L ("ST3 OUT") IS WIRED TO PIN LL ("SCHMITT RIG1"), AND "SCHMITT RIG 1" WILL SET CLOCK A'S STATUS REGISTER BIT 15.

IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIQUE. AND ERROR SWITCH REGISTER OPTIONS ARE USED. AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE TEST. SW13=1 WILL INHIBIT THIS FEATURE.

YOU MUST WIRE PINS V TO T OF J1 TOGETHER, AS WELL AS PINS L TO LL OF J1 TOGETHER.

TESTS LS210 AND LS214 SHOULD BE RUN FIRST.

9.2.4 LS224 "A EVENT OUT" TEST

THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND PROVIDING SCOPE LOOP CAPABILITIES FOR "A EVENT OUT".

WHEN YOU LOAD AND START AT LOCATION 224, PROGRAM CONTROL IS TRANSFERRED HERE. "A EVENT OUT" PULSES ARE GENERATED BY CLOCK A OVERFLOWS. PIN VV ("A EVENT OUT") IS WIRED TO PIN LL ("SCHMITT TRIG 1"). "SCHMITT TRIG 1" PULSES WILL SET CLOCK A'S CSR BIT 15. IF AN ERROR IS DETECTED, NORMAL ERROPORTING TECHNIQUE. AND ERROR SWITCH REGISTER OPTIONS ARE USED. AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE TEST. SW13=1 WILL INHIBIT THIS FEATURE.

YOU MUST WIRE PINS VV AND LL OF J1 TOGETHER.

TEST LS210 SHOULD BE RUN FIRST.

9.2.5 LS230 "B EVENT OUT" TEST

THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND PROVIDING SCOPE LOOP CAPABILITIES FOR "B EVENT OUT".

WHEN YOU LOAD AND START AT LOCATION 230, PROGRAM CONTROL IS TRANSFERRED HERE. "B EVENT OUT" PULSES ARE GENERATED BY CLOCK B OVERFLOWS. PIN TT ("B EVENT OUT") IS WIRED TO PIN LL ("SCHMITT TRIG 1"). "SCHMITT TRIG 1" PULSES WILL SET CLOCK A'S CSR BIT 15. AND ERROR SWITCH REGISTER OPTIONS ARE USED. AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE TEST. SW13=1 WILL INHIBIT THIS FEATURE.

YOU MUST WIRE PINS TT AND LL OF J1 TOGETHER.

TEST LS210 SHOULD BE RUN FIRST.

```
                        %

                .TITLE  MAINDEC-11-DZKWK-A
                ;*COPYRIGHT (C) 1976
                ;*DIGITAL EQUIPMENT CORP.
                ;*MAYNARD, MASS. 01754
                ;*
                ;*PROGRAM BY EDWARD C. BADGER
                ;*
                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                ;*PACKAGE (MAINDEC-11-DZQAC-B2),NOV 21, 1975.
                ;*
000001          $TN=1

                .SBTTL  OPERATIONAL SWITCH SETTINGS
                ;*
                ;*      SWITCH                  USE
                ;*      ------          --------------------
                ;*        15            HALT ON ERROR
                ;*        14            LOOP ON TEST
                ;*        13            INHIBIT ERROR TYPEOUTS
                ;*        11            INHIBIT ITERATIONS
                ;*        10            BELL ON ERROR
                ;*         9            LOOP ON ERROR
                ;*         8            LOOP ON TEST IN SWR<7:0>

                .SBTTL  TRAP CATCHER

000000                  .=0
                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                  .=174
```

```
                                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
                                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                         .=200
000200  000137  001634                  JMP     @#START         ;GO TO STARTING ADDRESS OF PROGRAM

000204  000137  002320                  JMP     @#RSTART        ;GO TO RESTART ADDRESS.
000210  000137  023446                  JMP     @#LS210         ;GO TO SPECIAL TEST #1.
000214  000137  023560                  JMP     @#LS214         ;GO TO SPECIAL TEST #2.
000220  000137  023704                  JMP     @#LS220         ;GO TO SPECIAL TEST #3.
000224  000137  024014                  JMP     @#LS224         ;GO TO SPECIAL TEST #4.
000230  000137  024142                  JMP     @#LS230         ;GO TO SPECIAL TEST #5.



                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
```

```
                        040000          SW14=   40000
                        020000          SW13=   20000
                        010000          SW12=   10000
                        004000          SW11=   4000
                        002000          SW10=   2000
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
```

```
                    000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                    000014          TBITVEC=14              ;;"T" BIT
                    000014          TRTVEC= 14              ;;TRACE TRAP
                    000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                    000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                    000024          PWRVEC= 24              ;;POWER FAIL
                    000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                    000034          TRAPVEC=34              ;;"TRAP" TRAP
                    000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
                    000064          TPVEC=  64              ;;TTY PRINTER VECTOR
                    000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                    170404                  ABASE=  170404
                    000344                  AVECT1= 344
                    000006                  APRIOR= 6



                                    .SBTTL  ACT11 HOOKS

                                    ;;**************************************************************
                                    ;HOOKS REQUIRED BY ACT11
                                            $SVPC=.                 ;SAVE PC
                                            .=46
                    000046  023414          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                                            .=52
                    000052  000000          .WORD   0               ;;2)SET LOC.52 TO ZERO
                                            .=$SVPC                 ;; RESTORE PC
                                            .=1000

                                    .SBTTL  APT PARAMETER BLOCK

                                    ;;**************************************************************
                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                    ;;**************************************************************
                                            .$X=.   ;;SAVE CURRENT LOCATION
                                            .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                    000024  000200          200     ;;FOR APT START UP
                                            .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                    000044  001000          $APTHDR ;;POINT TO APT HEADER BLOCK
                                            .=.$X   ;;RESET LOCATION COUNTER
                                    ;;**************************************************************
                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                    ;INTERFACE SPEC.

                    001000          $APTHD:
                    001000  000000  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
                    001002  001200  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
                    001004  000002  $TSTM:  .WORD   2       ;;RUN TIM OF LONGEST TEST
```

```
   919  001006  000120                  $PASTM: .WORD   120     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
   920  001010  000120                  $UNITM: .WORD   120     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
   921  001012  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
   922
```

```
                                        .SBTTL  COMMON TAGS

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

                001100                          .=1100
        001100                          $CMTAG:                         ;;START OF COMMON TAGS
        001100  000000                          .WORD   0
        001102     000                  $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
        001103     000                  $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
        001104  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
        001106  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
        001110  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
        001112  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
        001114     000                  $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
        001115     001                  $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
        001116  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
        001120  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
        001122  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
        001124  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
        001126  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
        001130  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
        001132  000000                          .WORD   0
        001134  000000                          .WORD   0
        001136  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
        001140  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
        001142  177560                  $TKS:   177560                  ;;TTY KBD STATUS
        001144  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
        001146  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
        001150  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
        001152     000                  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
        001153     002                  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
        001154     012                  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
        001155     000                  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
        001156  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                        ;;WHICH  ($REG0) WAS OBTAINED
        001160  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
        001162  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
        001164  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
        001166  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
        001170  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
        001174     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
        001175     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
        001176  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
```

```
                                ;;*****************************************************************

                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001200                          $MAIL:                          ;;APT MAILBOX
001200  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001202  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001204  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001206  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001210  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001212  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001214  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001216  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001220                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001220     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001221     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001222  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001224  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001226  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
                                ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,Q=10
                                ;*                              BIT 10=REAL TIME CLOCK
                                ;*                              BIT  9=FLOATING POINT PROCESSOR
                                ;*                              BIT  8=MEMORY MANAGEMENT
001230     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001231     000                  $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                                      900 NSEC CORE=001
                                ;*                                      300 NSEC BIPOLAR=002
                                ;*                                      500 NSEC MOS=003
001232  000000                  $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001234     000                  $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
001235     000                  $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
001236  000000                  $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
001240     000                  $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
001241     000                  $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
001242  000000                  $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
001244     000                  $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
001245     000                  $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
001246  000000                  $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
001250     344                  $VECT1: .BYTE   AVECT1          ;;INTERRUPT VECTOR#1
001251     000                  $VECT2: .BYTE   AVECT2          ;;INTERRUPT VECTOR#2
001252     006                  $PRIOR: .BYTE   APRIOR          ;;BUS PRIORITY #1,#2
001253     000                          .BYTE   0               ;;SPARE,NOT USED
                                        .EVEN
001254  170404                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001256  000000                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
001260  000000                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
001262  000000                  $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
001264  000000                  $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
001266  000000                  $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
001270  000000                  $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
```

```
1025  001272  000000          $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1026  001274  000000          $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1027  001276  000000          $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1028  001300  000000          $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1029  001302  000000          $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1030  001304  000000          $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
1031  001306  000000          $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
1032  001310  000000          $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
1033  001312  000000          $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
1034  001314  000000          $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
1035  001316  000000          $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
1036  001320  000000          $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
1037  001322  000000          $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
1038
1039
1040  001324                  $ETEND:
1041
1042
1043  001324  170404          ASR:    170404          ;/CLOCK A STATUS REGISTER.
1044  001326  170406          ABR:    170406          ;/CLOCK A BUFFER REGISTER.
1045  001330  170430          ACR:    170430          ;/CLOCK A COUNT REGISTER.
1046
1047  001332  170432          BSR:    170432          ;/CLOCK B STATUS REGISTER.
1048  001334  170434          BBR:    170434          ;/CLOCK B BUFFER REGISTER.
1049  001336  170436          BCR:    170436          ;/CLOCK B COUNT REGISTER.
1050
1051  001340  000344          AVECT:  344             ;/CLOCK A INTR. VECTOR ADDR.
1052  001342  000346          AVECP2: 346             ;/CLOCK A INTR. STATUS WORD.
1053
1054  001344  000364          BVECT:  364             ;/CLOCK B INTR. VECTOR ADDR.
1055  001346  000366          BVECT2: 366             ;/CLOCK B INTR. STATUS WORD.
1056
1057  001350  000006          APRITY: 6               ;/PRIORITY LEVEL OF CLOCK A.
1058  001352  000006          BPRITY: 6               ;/PRIORITY LEVEL OF CLOCK B.
1059
1060
1061
1062                          .SBTTL  ERROR POINTER TABLE
1063
1064                          ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1065                          ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1066                          ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1067                          ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1068                          ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1069
1070                          ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1071                          ;*      DH              ;;POINTS TO THE DATA HEADER
1072                          ;*      DT              ;;POINTS TO THE DATA
1073                          ;*      DF              ;;POINTS TO THE DATA FORMAT
1074
1075
1076  001354                  $ERRTB:
1077
1078                          ;ITEM   1
1079
1080  001354  027356                  EM1                     ;CLOCK A SR FUNCTION ERROR
```

```
1081  001356  030243                        DH1             ;ERRPC   ASR     WAS     S/B
1082  001360  031056                        DT1             ;$ERRPC,ASR,$BDDAT,$GDDAT
1083  001362  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1084
1085
1086                                ;ITEM   2
1087
1088  001364  027411                        EM2             ;CLOCKA SR DATA ERROR
1089  001366  030243                        DH1             ;ERRPC   ASR     WAS     S/B
1090  001370  031056                        DT1             ;$ERRPC,ASR,$BDDAT,$GDDAT
1091  001372  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1092
1093
1094                                ;ITEM   3
1095
1096  001374  027440                        EM3             ;CLOCKA BR DATA ERROR
1097  001376  030301                        DH3             ;ERRPC   ABR     WAS     S/B
1098  001400  031070                        DT3             ;$ERRPC,ABR,$BDDAT,$GDDAT
1099  001402  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1100
1101
1102                                ;ITEM   4
1103
1104  001404  027467                        EM4             ;CLOCKA CR DATA ERROR
1105  001406  030337                        DH4             ;ERRPC   ACR     WAS     S/B
1106  001410  031102                        DT4             ;$ERRPC,ACR,$BDDAT,$GDDAT
1107  001412  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1108
1109
1110                                ;ITEM   5
1111
1112  001414  027516                        EM5             ;CLOCK B SR DATA ERROR
1113  001416  030375                        DH5             ;ERRPC   BSR     WAS     S/B
1114  001420  031114                        DT5             ;$ERRPC,BSR,$BDDAT,$GDDAT
1115  001422  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1116
1117
1118                                ;ITEM   6
1119
1120  001424  027545                        EM6             ;CLOCK B BR DATA ERROR
1121  001426  030433                        DH6             ;ERRPC   BBR     WAS     S/B
1122  001430  031126                        DT6             ;$ERRPC,BBR,$BDDAT,$GDDAT
1123  001432  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1124
1125
1126                                ;ITEM   7
1127
1128  001434  027574                        EM7             ;CLOCK B CR DATA ERROR
1129  001436  030471                        DH7             ;ERRPC   BCR     WAS     S/B
1130  001440  031140                        DT7             ;$ERRPC,BCR,$BDDAT,$GDDAT
1131  001442  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1132
1133
1134                                ;ITEM   10
1135
1136  001444  027623                        EM10            ;DUAL ADDRESS ERROR
```

```
1137  001446  030527                        DH10            ;ERROR    GOOD    BAD     GOOD    DATA READ FROM
1138                                                        ;  PC     ADDR    ADDR    DATA    DUAL ADDRESS
1139  001450  031152                        DT10            ;$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT
1140  001452  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1141
1142
1143                              ;ITEM   11
1144
1145  001454  027650                        EM11            ;CLOCK A COUNT ERROR
1146  001456  030337                        DH4             ;ERRPC   ACR   WAS   S/B
1147  001460  031102                        DT4             ;$ERRPC, ACR, $BDDAT, $GDDAT
1148  001462  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1149
1150
1151                              ;ITEM   12
1152
1153  001464  027677                        EM12            ;CLOCK A COUNT FUNCTION ERROR
1154  001466  030666                        DH12            ;ERRPC   ASR
1155  001470  031166                        DT12            ;ERRPC, ASR
1156  001472  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1157
1158
1159                              ;ITEM   13
1160
1161  001474  031260                        DF0             ;ERROR 13 DOES NOT EXSIST.
1162  001476  031260                        DF0             ;IT WOULD BE BAD LUCK.
1163  001500  031260                        DF0
1164  001502  031260                        DF0
1165
1166                              ;ITEM   14
1167
1168  001504  027737                        EM14            ;CLOCK B COUNT FUNCTION ERROR
1169  001506  030705                        DH14            ;ERRPC   BSR
1170  001510  031174                        DT14            ;$ERRPC, BSR
1171  001512  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1172
1173
1174                              ;ITEM   15
1175
1176  001514  027777                        EM15            ;CLOCK B COUNT ERROR
1177  001516  030471                        DH7             ;ERRPC   CSR   WAS   S/B
1178  001520  031140                        DT7             ;$ERRPC, BCR, $BDDAT, $GDDAT
1179  001522  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1180
1181
1182                              ;ITEM   16
1183
1184  001524  030026                        EM16            ;CLOCK A INTERRUPT ERROR
1185  001526  030666                        DH12            ;ERRPC   ASR
1186  001530  031166                        DT12            ;$ERRPC, ASR
1187  001532  031260                        DF0             ;ALL NUMBERS ARE IN OCTAL FORM
1188
1189
1190                              ;ITEM   17
1191
1192  001534  030061                        EM17            ;CLOCK B INTERRUPT ERROR
```

```
1193  001536  030705                    DH14                ;ERRPC   BSR
1194  001540  031174                    DT14                ;$ERRPC, BSR
1195  001542  031260                    DF0
1196
1197                              ;ITEM   20
1198
1199  001544  030114                    EM20                ;CLOCK A REPEATABILITY ERROR
1200  001546  030723                    DH20                ;ERROR   ASR   2ND CNT 1ST CNT
1201  001550  031056                    DT1                 ;$ERRPC, ASR, $BDDAT, $GDDAT
1202  001552  031260                    DF0                 ;ALL NUMBERS ARE IN OCTAL FORM
1203
1204
1205                              ;ITEM   21
1206
1207  001554  027650                    EM11                ;CLOCK A COUNT ERROR
1208  001556  030337                    DH4                 ;ERROR   ASR   2ND CNT 1ST CNT
1209  001560  031202                    DT21                ;$ERRPC, ASR, $BDDAT, $GDDAT
1210  001562  031260                    DF0                 ;ALL NUMBERS ARE IN OCTAL FORM
1211
1212
1213                              ;ITEM   22
1214
1215  001564  027650                    EM11                ;CLOCK A COUNT ERROR
1216  001566  030337                    DH4                 ;ERRPC   ASR   WAS   S/B
1217  001570  031214                    DT22                ;$ERRPC, ACR, $BDDAT, $TMPO
1218  001572  031260                    DF0                 ;ALL NUMBERS ARE IN OCTAL FORM
1219
1220
1221                              ;ITEM   23
1222
1223  001574  030153                    EM23                ;CLOCK B REPEATABILITY ERROR
1224  001576  030765                    DH23                ;ERROR   ASR   2NDCNT  1STCNT
1225  001600  031056                    DT1                 ;$ERRPC, ASR, $BDDAT, $GDDAT
1226  001602  031260                    DF0                 ;ALL NUMBERS ARE IN OCTAL FORM
1227
1228
1229                              ;ITEM   24
1230
1231  001604  027777                    EM15                ;CLOCK B COUNT ERROR
1232  001606  030471                    DH7                 ;ERRPC   BCR   WAS   S/B
1233  001610  031226                    DT24                ;$ERRPC, BCR, $GDDAT, $TMPO
1234  001612  031260                    DF0                 ;ALL NUMBERS ARE IN OCTAL FORM
1235
1236
1237                              ;ITEM   25
1238
1239  001614  027777                    EM15                ;CLOCK B COUNT ERROR
1240  001616  030471                    DH7                 ;ERRPC   BCR     WAS     S/B
1241  001620  031240                    DT25                ;$ERRPC,BCR,$BDDAT,$TMPO
1242  001622  031260                    DF0                 ;ALL NUMBERS ARE IN OCTAL FORM
1243
1244
1245                              ;ITEM   26
1246
1247  001624  030212                    EM26    ;CLOCK ADDRESSING ERRROR
1248  001626  031027                    DH26    ;ERRPC   CLOCK ADDR.
```

```
1249  001630  031252                          D[26    ;$ERRPC,$TMPO
1250  001632  031260                          DFD                    ;ALL NUMBERS ARE IN OCTAL FORM
1251
1252
1253
1254                                          .SBTTL  PROGRAM START
```

```
                                        START:
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001634                                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001634  012706  001100                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001640  005026                                  CMP     #$BDDAT,R6      ;;DONE?
001642  022706  001126                          BNE     .-6             ;;LOOP BACK IF NO
001646  001374                                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
001650  012706  001100
                                        ;;INITIALIZE A FEW VECTORS
001654  012737  025434  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001662  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001670  012737  024672  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001676  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001704  012737  027312  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001712  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
001720  012737  027134  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001726  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001734  005037  001164                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001740  005037  001166                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001744  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
001752  012737  001752  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001760  012737  001760  001110                  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001766  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001772  012737  002030  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002000  012737  177570  001136                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002006  012737  177570  001140                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002014  022777  177777  177114                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002022  001013                                  BNE     65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
002024  005737  000001                          TST     @#1             ;;FORCE A TRAP THROUGH ERRVEC
002030  012737  000176  001136  64$:            MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002036  012737  000174  001140                  MOV     #DISPREG,DISPLAY;;POINT TO SOFTWARE DISPLAY REG
002044  012716  002052                          MOV     #65$,(SP)       ;;REPLACE OLD PC WITH NEW
002050  000002                                  RTI                     ;;RESTORE PC AND PSW
002052  012637  000004          65$:            MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002056                          $ARG1:
002056  005037  001206                          CLR     $PASS           ;;CLEAR PASS COUNT
002062  132737  000200  001221                  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002070  001403                                  BEQ     64$             ;;YES, USE NON-APT SWITCH
002072  012737  001222  001136                  MOV     #$SWREG,SWR     ;;NO, USE APT SWITCH REGISTER
002100                          64$:
002100  005737  000042                          TST     @#42            ;IF RUNNING UNDER ACT-
002104  001015                                  BNE     10$             ;NO TYPEOUT.

002106  104400  002114                          TYPE    ,66$            ;;TYPE ASCIZ STRING
002112  000412                                  BR      65$             ;;GET OVER THE ASCIZ

002140                          ;;66$:  .ASCIZ  <15><12><12>#MD-11-DZKWK-A#<15><12>
                                65$:

002140  013737  001254  001324  10$:    MOV     $BASE,ASR
002146  013737  001250  001340          MOV     $VECT1,AVECT
002154  013737  001252  001350          MOV     $PRIOR,APRITY
002162  012737  000001  001210          MOV     #1,$DEVCT
```

```
1311  002170  005037  001206                    CLR     $PASS
1312
1313  002174                            LOOP:
1314  002174  005000                            CLR     R0
1315  002176  005200                    1$:     INC     R0              ;DELAY SOME TIME SO THAT FIRST RESET
1316  002200  001376                            BNE     1$              ;INSTR. WON'T CLOBBER TYPEOUT.
1317  002202  013700  001324                    MOV     ASR,R0          ;NOW WE'RE GONNA FIX
1318  002206  062700  000002                    ADD     #2,R0           ;ALL CLOCK ADDRESSES BASED ON ASR.
1319  002212  010037  001326                    MOV     R0,ABR
1320  002216  062700  000022                    ADD     #22,R0
1321  002222  010037  001330                    MOV     R0,ACR
1322  002226  062700  000002                    ADD     #2,R0
1323  002232  010037  001332                    MOV     R0,BSR
1324  002236  062700  000002                    ADD     #2,R0
1325  002242  010037  001334                    MOV     R0,BBR
1326  002246  062700  000002                    ADD     #2,R0
1327  002252  010037  001336                    MOV     R0,BCR
1328
1329  002256  013700  001340                    MOV     AVECT,R0        ;NOW FIX VECTOR ADDRESSES
1330  002262  062700  000002                    ADD     #2,R0           ;BASED ON AVECT.
1331  002266  010037  001342                    MOV     R0,AVECP2
1332  002272  062700  000016                    ADD     #16,R0
1333  002276  010037  001344                    MOV     R0,BVECT
1334  002302  062700  000002                    ADD     #2,R0
1335  002306  010037  001346                    MOV     R0,BVECT2
1336
1337  002312  013737  001350  001352            MOV     APRITY,BPRITY   ;FIX CLK B'S PRIORITY BASED ON A'S.
1338  002320  012706  001100            RSTART: MOV     #STACK,SP
1339  002324  012746  000340                    MOV     #340,-(SP)      ;SET PROCESSOR PRIORITY TO 7.
1340  002330  012746  002336                    MOV     #1$,-(SP)
1341  002334  000002                            RTI
1342  002336                            1$:
1343
1344                                    .SBTTL  *
1345                                    .SBTTL  * PHASE 1 CLOCKS A+B BASIC LOGIC TESTS.
1346                                    .SBTTL  *
```

```
                                        ;;**********************************************************************
                                        ;*TEST 1         *TEST THE ADDRESSABILITY OF CLOCK A'S CSR
                                        ;*
                                        ;*"BUS A17":"A04"="DEVICE" H; "DEVICE" H +"TP0" H='DEV ENABLE" H
                                        ;*"DEV ENABLE" H+"TP1" H="DEV ENB 2" H
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17"
                                        ;*
                                        ;* CLOCK ADDRESS  TEST. SCOPE FOR "DEV ENB Z" H AND WORK BACK
                                        ;*
                                        ;**********************************************************************
002336  000240                TST1:   NOP
002340  012737  000050  001164        MOV     #50,$TIMES      ;;DO 50 ITERATIONS
002346  012737  002372  001106        MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
002354  112737  000001  001102        MOVB    #1,$TSTNM                       ;/THIS IS THE FIRST TEST.

002362  005037  001124                CLR     $GDDAT          ;CLEAR THESE LOCATIONS ON START OF PROG.
002366  005037  001126                CLR     $BDDAT          ;MAKES FIRST ERROR CLEAN TYPEOUT.
002372  013746  000004        1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
002376  012737  002412  000004        MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
                                                            ;WE TIME-OUT WHEN ADDRESSING THE KW11.
002404  005777  176714                TST     @ASR            ;ADDRESS THE CLOCK!
                                                            ;IF CLOCK DOES NOT RETURN
                                                            ;"BUS SSYN" THEN WE'LL TIME-OUT.
002410  000406                        BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
002412                      2$:
002412  062706  000004                ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
002416  013737  001324  001162        MOV     ASR,$TMP0       ;FOR ERROR TYPEOUT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

002424  104026                        ERROR   26              ;REPORT ERROR=CLOCK A'S CSR FAILED TO RETURN
                                                            ;"BUS SSYN" WHEN ADDRESSED.
                                                            ;NOTE: IF PROGRAM HAS INCORRECT
                                                            ;ADDRESS THEN WE MIGHT NOT BE
                                                            ;TALKING TO THE CLOCK. MAKE SURE
                                                            ;OF CLOCK ADDRESS.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

002426  012637  000004        3$:     MOV     (SP)+,@#ERRVEC
```

```
                                        ;;**********************************************************************
                                        ;*TEST 2        *TEST THE ADDRESSABILITY OF CLOCK A'S BUFFER REG.
                                        ;*
                                        ;*"BUS A17":"A04"="DEVICE" H; "DEVICE" H +"TP0" H="DEV ENABLE" H
                                        ;*"DEV ENABLE" H+"TP1" H="DEV ENB 2" H
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17"
                                        ;*
                                        ;* CLOCK ADDRESS  TEST. SCOPE FOR "DEV ENB Z" H AND WORK BACK
                                        ;*
                                        ;**********************************************************************
002432  000004                  TST2:   SCOPE

002434  005037  001124                  CLR     $GDDAT          ;CLEAR THESE LOCATIONS ON START OF PROG.
002440  005037  001126                  CLR     $BDDAT          ;MAKES FIRST ERROR CLEAN TYPEOUT.
002444  013746  000004          1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
002450  012737  002464  000004        MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
                                                                ;WE TIME-OUT WHEN ADDRESSING THE KW11.
002456  005777  176644                  TST     @ABR            ;ADDRESS THE CLOCK!
                                                                ;IF CLOCK DOES NOT RETURN
                                                                ;"BUS SSYN" THEN WE'LL TIME-OUT.
002462  000406                        BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
002464                          2$:
002464  062706  000004                ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
002470  013737  001326  001162        MOV     ABR,$TMP0       ;FOR ERROR TYPEOUT.

                ;;;,SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

002476  104026                        ERROR   26              ;REPORT ERROR=CLOCK A'S BUFFER REG. FAILED TO RETURN
                                                                ;"BUS SSYN" WHEN ADDRESSED.
                                                                ;NOTE: IF PROGRAM HAS INCORRECT
                                                                ;ADDRESS THEN WE MIGHT NOT BE
                                                                ;TALKING TO THE CLOCK. MAKE SURE
                                                                ;OF CLOCK ADDRESS.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

002500  012637  000004          3$:     MOV     (SP)+,@#ERRVEC
```

```
                                        ;;**********************************************************************
                                        ;*TEST 3        *TEST THE ADDRESSABILITY OF CLOCK A'S COUNT  REG.
                                        ;*
                                        ;*"BUS A17":"A04"="DEVICE" H; "DEVICE" H +"TP0" H="DEV ENABLE" H
                                        ;*"DEV ENABLE" H+"TP1" H="DEV ENB 2" H
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17"
                                        ;*
                                        ;* CLOCK ADDRESS  TEST. SCOPE FOR "DEV ENB Z" H AND WORK BACK
                                        ;*
                                        ;;**********************************************************************
002504  000004                        TST3:   SCOPE

002506  005037  001124                          CLR     $GDDAT          ;CLEAR THESE LOCATIONS ON START OF PROG.
002512  005037  001126                          CLR     $BDDAT          ;MAKES FIRST ERROR CLEAN TYPEOUT.
002516  013746  000004                  1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
002522  012737  002536  000004                  MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
                                                                        ;WE TIME-OUT WHEN ADDRESSING THE KW11.

002530  005777  176574                          TST     @ACR            ;ADDRESS THE CLOCK!
                                                                        ;IF CLOCK DOES NOT RETURN
                                                                        ;"BUS SSYN" THEN WE'LL TIME-OUT.
002534  000406                          BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
002536                          2$:
002536  062706  000004                  ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
002542  013737  001330  001162          MOV     ACR,$TMP0       ;FOR ERROR TYPEOUT.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

002550  104026                          ERROR   26              ;REPORT ERROR=CLOCK A'S COUNT  REG. FAILED TO RETURN
                                                                ;"BUS SSYN" WHEN ADDRESSED.
                                                                ;NOTE: IF PROGRAM HAS INCORRECT
                                                                ;ADDRESS THEN WE MIGHT NOT BE
                                                                ;TALKING TO THE CLOCK. MAKE SURE
                                                                ;OF CLOCK ADDRESS.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

002552  012637  000004          3$:     MOV     (SP)+,@#ERRVEC
```

```
1491
1492                                    ;;*****************************************************************
1493                                    ;*TEST 4        *TEST THE ADDRESSABILITY OF CLOCK B'S CSR
1494                                    ;*
1495                                    ;*"BUS A17":"A04"="DEVICE" H; "DEVICE" H +"TP0" H="DEV ENABLE" H
1496                                    ;*"DEV ENABLE" H+"TP1" H="DEV ENB 2" H
1497                                    ;*
1498                                    ;*
1499                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17"
1500                                    ;*
1501                                    ;* CLOCK ADDRESS  TEST. SCOPE FOR "DEV ENB Z" H AND WORK BACK
1502                                    ;*
1503                                    ;*****************************************************************
1504  002556  000004                    TST4:   SCOPE
1505
1506  002560  005037  001124                    CLR     $GDDAT          ;CLEAR THESE LOCATIONS ON START OF PROG.
1507  002564  005037  001126                    CLR     $BDDAT          ;MAKES FIRST ERROR CLEAN TYPEOUT.
1508  002570  013746  000004            1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
1509  002574  012737  002610  000004            MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
1510                                                                    ;WE TIME-OUT WHEN ADDRESSING THE KW11.
1511
1512  002602  005777  176524                    TST     @BSR            ;ADDRESS THE CLOCK!
1513                                                                    ;IF CLOCK DOES NOT RETURN
1514                                                                    ;"BUS SSYN" THEN WE'LL TIME-OUT.
1515  002606  000406                            BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
1516  002610                            2$:
1517  002610  062706  000004                    ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
1518  002614  013737  001332  001162            MOV     BSR,$TMP0       ;FOR ERROR TYPEOUT.
1519
1520
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

1524  002622  104026                            ERROR   26              ;REPORT ERROR=CLOCK B'S CSR FAILED TO RETURN
1525                                                                    ;"BUS SSYN" WHEN ADDRESSED.
1526                                                                    ;NOTE: IF PROGRAM HAS INCORRECT
1527                                                                    ;ADDRESS THEN WE MIGHT NOT BE
1528                                                                    ;TALKING TO THE CLOCK. MAKE SURE
1529                                                                    ;OF CLOCK ADDRESS.
1530
1531
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

1535  002624  012637  000004            3$:     MOV     (SP)+,@#ERRVEC
1536
```

```
1537
1538                                    ;;*********************************************************************
1539                                    ;*TEST 5        *TEST THE ADDRESSABILITY OF CLOCK B'S BUFFER REG.
1540                                    ;*
1541                                    ;*"BUS A17":"A04"="DEVICE" H; "DEVICE" H +"TP0" H="DEV ENABLE" H
1542                                    ;*"DEV ENABLE" H+"TP1" H="DEV ENB 2" H
1543                                    ;*
1544                                    ;*
1545                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17"
1546                                    ;*
1547                                    ;* CLOCK ADDRESS  TEST. SCOPE FOR "DEV ENB 2" H AND WORK BACK
1548                                    ;*
1549                                    ;*********************************************************************
1550  002630  000004                    TST5:   SCOPE
1551
1552  002632  005037  001124                    CLR     $GDDAT          ;CLEAR THESE LOCATIONS ON START OF PROG.
1553  002636  005037  001126                    CLR     $BDDAT          ;MAKES FIRST ERROR CLEAN TYPEOUT.
1554  002642  013746  000004            1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
1555  002646  012737  002662  000004            MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
1556                                                                    ;WE TIME-OUT WHEN ADDRESSING THE KW11.
1557
1558  002654  005777  176454                    TST     @38R            ;ADDRESS THE CLOCK!
1559                                                                    ;IF CLOCK DOES NOT RETURN
1560                                                                    ;"BUS SSYN" THEN WE'LL TIME-OUT.
1561  002660  000406                            BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
1562  002662                            2$:
1563  002662  062706  000004                    ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
1564  002666  013737  001334  001162            MOV     @R6,$TMP0       ;FOR ERROR TYPEOUT.
1565
1566

            ;;;SSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSS

1570  002674  104026                            ERROR   26              ;REPORT ERROR=CLOCK B'S BUFFER REG. FAILED TO RETURN
1571                                                                    ;"BUS SSYN" WHEN ADDRESSED.
1572                                                                    ;NOTE: IF PROGRAM HAS INCORRECT
1573                                                                    ;ADDRESS THEN WE MIGHT NOT BE
1574                                                                    ;TALKING TO THE CLOCK. MAKE SURE
1575                                                                    ;OF CLOCK ADDRESS.
1576
1577

            ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

1581  002676  012637  000004            3$:     MOV     (SP)+,@#ERRVEC
1582
```

```
1583
1584                                    ;;*****************************************************************
1585                                    ;*TEST 6        *TEST THE ADDRESSABILITY OF CLOCK B'S COUNT  REG.
1586                                    ;*
1587                                    ;*"BUS A17":"A04"="DEVICE" H; "DEVICE" H +"TP0" H="DEV ENABLE" H
1588                                    ;*"DEV ENABLE" H+"TP1" H="DEV ENB 2" H
1589                                    ;*
1590                                    ;*
1591                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17"
1592                                    ;*
1593                                    ;* CLOCK ADDRESS  TEST. SCOPE FOR "DEV ENB Z" H AND WORK BACK
1594                                    ;*
1595                                    ;*****************************************************************
1596  002702  000004                    TST6:   SCOPE
1597
1598  002704  005037  001124                    CLR     $GDDAT          ;CLEAR THESE LOCATIONS ON START OF PROG.
1599  002710  005037  001126                    CLR     $BDDAT          ;MAKES FIRST ERROR CLEAN TYPEOUT.
1600  002714  013746  000004            1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
1601  002720  012737  002734  000004            MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
1602                                                                    ;WE TIME-OUT WHEN ADDRESSING THE KW11.
1603
1604  002726  005777  176404                    TST     @BCR            ;ADDRESS THE CLOCK!
1605                                                                    ;IF CLOCK DOES NOT RETURN
1606                                                                    ;"BUS SSYN" THEN WE'LL TIME-OUT.
1607  002732  000406                            BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
1608  002734                            2$:
1609  002734  062706  000004                    ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
1610  002740  013737  001336  001162            MOV     BCR,$TMP0       ;FOR ERROR TYPEOUT.
1611
1612

            ;;;$$$
```

```
                                ;;**********************************************************************
                                ;*TEST 7        *TEST THAT CLOCK A BUFFER CAN BE WRITTEN INTO
                                ;*
                                ;*FOR LOADING DATA:
                                ;*
                                ;*(WE KNOW WE CAN ADDR. KW11), "BUS A01" L + "A02" H + "A03" H
                                ;*+"BUS C1" L="LD BUFF A" L
                                ;*"LD BUF A" L + BUFFERED DATA LOADS INTO MUX LATCH (NOTE WE KNOW
                                ;*BY NOW "TP1" L SHOULD BE GOOD).
                                ;*
                                ;* FOR READING DATA:
                                ;*
                                ;*BUS A01 L + (DATA IN H + EV ENABLE(1) H)=RD BUFF AL
                                ;*[BA01H*(DEPENDING ON WHICH DATA BITS READ) RD BUF AL]+BUFF A00:15
                                ;* +[DEV ENABLE*DATA IN L]=BUS DATA
                                ;*
                                ;*SINCE WE WONT LOOK FOR ANY SPECIFIC DATA BIT FAILURE,
                                ;*JUST THAT WE CAN WRITE INTO BUFFER + READ BACK,
                                ;*IF FAILED, KEY ON "LD BUFF A L" AND "RD BUFF A'L"
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "BUS A17" (2 OCCURANCES PER LOOP)
                                ;*
                                ;*
                                ;;**********************************************************************
002754  000004                TST7:   SCOPE
002756  012737  000050  001164          MOV     #50,$TIMES      ;;DO 50 ITERATIONS

002764  012777  001416  176334          MOV     #1416,@ABR      ;WRITE INTO PRESET BUFFER.
002772  012737  176362  001124          MOV     #-1416,$GDDAT   ;SET $GDDAT FOR ERROR TYPEOUT.
003000  017737  176322  001126          MOV     @ABR,$BDDAT     ;READ THE PRESET BUFFER.
003006  001001                          BNE     1$              ;IF ANY DATA WAS READ BACK, WE WILL
                                                                ;BR PAST THE ERROR CALL.


                        ;;;SSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSS

003010  104003                          ERROR   3               ;UNABLE TO LOAD AND READ BACK
                                                                ;FROM THE BUFFER REGISTER CLOCK A.

                        ;;;SSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSS

003012                  1$:
```

```
1676
1677                                    ;;*****************************************************************
1678                                    ;*TEST 10       *TEST THAT CLOCK A BUFFER CAN BE WRITTEN TO A ZERO
1679                                    ;*
1680                                    ;*THE LAST TEST WROTE 1'S INTO CLOCK A'S BUFFER. IN THIS
1691                                    ;*TEST WE TRY TO WRITE ALL ZEROS.
1682                                    ;*
1683                                    ;*SIGNALS - SAME AS LAST TEST. SUSPECT F/F OR DATA GATE STUCK OPEN
1684                                    ;*
1685                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
1686                                    ;*
1687                                    ;*
1688
1689                                    ;;*****************************************************************
1690  003012  000004                    TST10:  SCOPE
1691  003014  012737  000050  001164            MOV     #50,$TIMES      ;;DO 50 ITERATIONS
1692
1693  003022  005037  001124            CLR     $GDDAT          ;INDICATE WE EXPECT 0'S.
1694  003026  005077  176274            CLR     @ABR            ;CLEAR BUFFER REGISTER.
1695  003032  017737  176270  001126    MOV     @ABR,$BDDAT     ;READ CLOCK A'S BUFFER REGISTER
1696  003040  001401                    BEQ     1$              ;SHOULD BE CLEAR - IF CLEAR - NEXT TEST.
1697
1698
            ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

1702  003042  104003                    ERROR   3               ;CLEAR INTR. FAILED TO CLEAR CLOCK A'S
1703                                                            ;BUFFER REGISTER.
1704
            ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

1708  003044                            1$:
1709
1710
1711                                    ;;*****************************************************************
1712                                    ;*TEST 11       *TEST THAT CLOCK A'S STATUS CAN BE WRITTEN AND READ
1713                                    ;*
1714                                    ;*NOW THAT WE CAN WRITE INTO THE BUFFER REGISTER, WE'RE GOING TO TRY
1715                                    ;*WRITING INTO THE STATUS REGISTER AND READ IT BACK.
1716                                    ;*
1717                                    ;*NEW SIGNALS: ["BA03" L + "BA02" L + "BA01" L]="LD STAT A" L
1718                                    ;*      "DATA OUT LO" L + "DATA OUT HI" L + "LD STATA" L = "LD STAT A HI" H
1719                                    ;*      + "LD STATA LO H"
1720                                    ;*FOR READ BACK: "RD STATA" L
1721                                    ;*
1722                                    ;*NO ATTEMPT MADE TO VERIFY THAT CORRECT DATA CAME BACK, BUT
1723                                    ;*JUST THAT SOME DATA CAME BACK.
1724                                    ;*
1725                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEV ENABLE (1)" 2 OCCURANCES PER PASS
1726                                    ;*
1727                                    ;*
1728
1729                                    ;;*****************************************************************
1730  003044  000004                    TST11:  SCOPE
1731  003046  012737  000050  001164            MOV     #50,$TIMES      ;;DO 50 ITERATIONS
```

```
1732
1733  003054  012737  001416  001124          MOV     #1416,$GDDAT    ;LOAD $GDDAT WITH S/B.
1734  003062  013777  001124  176234          MOV     $GDDAT,@ASR     ;LOAD STATUS REGISTER.
1735  003070  017737  176230  001126          MOV     @ASR,$BDDAT     ;READ BACK STATUS REGISTER
1736  003076  001001                          BNE     1$              ;IF ANY BITS RETURNED - NO ERROR.
1737

                ;;;$$$
```

```
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEV ENABLE (1)" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;**************************************************************************
003144  000004                  TST13:  SCOPE
003146  012737  000050  001164          MOV     #50,$TIMES      ;;DO 50 ITERATIONS

003154  012737  000370  001124          MOV     #370,$GDDAT     ;USE PATTERN "370", PUT IN $GDDAT.
003162  013777  001124  176144          MOV     $GDDAT,@BBR     ;WRITE INTO CLOCK B'S BUFFER REGISTER.
003170  017737  176140  001126          MOV     @BBR,$BDDAT     ;READ IT BACK.
003176  001001                          BNE     1$              ;IF ANY BITS COME BACK-SUBTEST OK.

        ;;;$$$
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 14        *TEST THAT CLOCK A STATUS REGISTER BIT 15 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
003202  000004                TST14:  SCOPE
003204  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003212  005077  176106                  CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003216  052777  100000  176100          BIS     #BIT15,@ASR     ;/SET BIT 15.
003224  012737  100000  001124          MOV     #BIT15,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
003232  017737  176066  001126          MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003240  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 15 AND ONLY BIT 15 SET?
003246  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003250  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 15 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003252  000412                          BR      2$              ;/BR TO END SUBTEST.
003254  042777  100000  176042  1$:     BIC     #BIT15,@ASR     ;/TRY CLEARING BIT 15.
003262  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
003266  017737  176032  001126          MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003274  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003276  104002                          ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 15 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003300                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 15       *TEST THAT CLOCK A STATUS REGISTER BIT 14 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003300  000004                  TST15:  SCOPE
003302  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003310  005077  176010                  CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003314  052777  040000  176002          BIS     #BIT14,@ASR     ;/SET BIT 14.
003322  012737  040000  001124          MOV     #BIT14,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
003330  017737  175770  001126          MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003336  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 14 AND ONLY BIT 14 SET?
003344  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

003346  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 14 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

003350  000412                          BR      2$              ;/BR TO END SUBTEST.
003352  042777  040000  175744  1$:     BIC     #BIT14,@ASR     ;/TRY CLEARING BIT 14.
003360  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
003364  017737  175734  001126          MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003372  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

003374  104002                          ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 14 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

003376                          2$:
```

```
                                        ;/*
                                        ;;*********************************************************************
                                        ;*TEST 16       *TEST THAT CLOCK A STATUS REGISTER BIT 13 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*********************************************************************
003376  000004                    TST16:  SCOPE
003400  012737  000100  001164            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003406  005077  175712                    CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003412  052777  020000  175704            BIS     #BIT13,@ASR     ;/SET BIT 13.
003420  012737  020000  001124            MOV     #BIT13,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
003426  017737  175672  001126            MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003434  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 13 AND ONLY BIT 13 SET?
003442  001402                            BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003444  104002                            ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                    ;/BIT 13 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003446  000412                            BR      2$              ;/BR TO END SUBTEST.
003450  042777  020000  175646  1$:       BIC     #BIT13,@ASR     ;/TRY CLEARING BIT 13.
003456  005037  001124                    CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
003462  017737  175636  001126            MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003470  001401                            BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003472  104002                            ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                    ;/BIT 13 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003474                          2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 17        *TEST THAT CLOCK A STATUS REGISTER BIT 9 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
003474  000004                 TST17:  SCOPE
003476  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003504  005077  175614                  CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003510  052777  001000  175606          BIS     #BIT9,@ASR      ;/SET BIT 9.
003516  012737  001000  001124          MOV     #BIT9,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003524  017737  175574  001126          MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003532  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 9 AND ONLY BIT 9 SET?
003540  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003542  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 9 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSSS

003544  000412                          BR      2$              ;/BR TO END SUBTEST.
003546  042777  001000  175550  1$:     BIC     #BIT9,@ASR      ;/TRY CLEARING BIT 9.
003554  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003560  017737  175540  001126          MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003566  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSSS

003570  104002                          ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 9 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSSS

003572                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 20       *TEST THAT CLOCK A STATUS REGISTER BIT 8 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003572  000004                          TST20:  SCOPE
003574  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003602  005077  175516                          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003606  052777  000400  175510                  BIS     #BIT8,@ASR      ;/SET BIT 8.
003614  012737  000400  001124                  MOV     #BIT8,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003622  017737  175476  001126                  MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003630  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 8 AND ONLY BIT 8 SET?
003636  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

003640  104002                                  ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 8 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

003642  000412                                  BR      2$              ;/BR TO END SUBTEST.
003644  042777  000400  175452  1$:             BIC     #BIT8,@ASR      ;/TRY CLEARING BIT 8.
003652  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
003656  017737  175442  001126                  MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003664  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

003666  104002                                  ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 8 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

003670                          2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 21        *TEST THAT CLOCK A STATUS REGISTER BIT 7 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
003670  000004                 TST21:  SCOPE
003672  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003700  005077  175420                 CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003704  052777  000200  175412         BIS     #BIT7,@ASR      ;/SET BIT 7.
003712  012737  000200  001124         MOV     #BIT7,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003720  017737  175400  001126         MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003726  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
003734  001402                       BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003736  104002                       ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                             ;/BIT 7 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003740  000412                       BR      2$              ;/BR TO END SUBTEST.
003742  042777  000200  175354  1$:  BIC     #BIT7,@ASR      ;/TRY CLEARING BIT 7.
003750  005037  001124                 CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003754  017737  175344  001126         MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003762  001401                       BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003764  104002                       ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                             ;/BIT 7 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

003766                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 22       *TEST THAT CLOCK A STATUS REGISTER BIT 6 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003766  000004                    TST22:  SCOPE
003770  012737  000100  001164            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003776  005077  175322                    CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
004002  052777  000100  175314            BIS     #BIT6,@ASR      ;/SET BIT 6.
004010  012737  000100  001124            MOV     #BIT6,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004016  017737  175302  001126            MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
004024  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
004032  001402                            BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

004034  104002                            ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                    ;/BIT 6 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

004036  000412                            BR      2$              ;/BR TO END SUBTEST.
004040  042777  000100  175256  1$:       BIC     #BIT6,@ASR      ;/TRY CLEARING BIT 6.
004046  005037  001124                    CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
004052  017737  175246  001126            MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
004060  001401                            BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

004062  104002                            ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                    ;/BIT 6 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

004064                          2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 23        *TEST THAT CLOCK A STATUS REGISTER BIT 5 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
004064  000004                          TST23:  SCOPE
004066  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004074  005077  175224                          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
004100  052777  000040  175216                  BIS     #BIT5,@ASR      ;/SET BIT 5.
004106  012737  000040  001124                  MOV     #BIT5,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004114  017737  175204  001126                  MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
004122  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
004130  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004132  104002                                  ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 5 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004134  000412                                  BR      2$              ;/BR TO END SUBTEST.
004136  042777  000040  175160  1$:             BIC     #BIT5,@ASR      ;/TRY CLEARING BIT 5.
004144  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
004150  017737  175150  001126                  MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
004156  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004160  104002                                  ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 5 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004162                                  2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 24        *TEST THAT CLOCK A STATUS REGISTER BIT 3 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
004162  000004                          TST24:  SCOPE
004164  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004172  005077  175126                          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
004176  052777  000010  175120                  BIS     #BIT3,@ASR      ;/SET BIT 3.
004204  012737  000010  001124                  MOV     #BIT3,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004212  017737  175106  001126                  MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
004220  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
004226  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004230  104002                                  ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 3 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004232  000412                                  BR      2$              ;/BR TO END SUBTEST.
004234  042777  000010  175062  1$:             BIC     #BIT3,@ASR      ;/TRY CLEARING BIT 3.
004242  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
004246  017737  175052  001126                  MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
004254  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004256  104002                                  ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 3 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

004260                                  2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 25        *TEST THAT CLOCK A STATUS REGISTER BIT 2 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
004260  000004                TST25:  SCOPE
004262  012737  000100  001164        MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004270  005077  175030              CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
004274  052777  000004  175022      BIS     #BIT2,@ASR      ;/SET BIT 2.
004302  012737  000004  001124      MOV     #BIT2,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004310  017737  175010  001126      MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
004316  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
004324  001402                      BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004326  104002                      ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                            ;/BIT 2 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004330  000412                      BR      2$              ;/BR TO END SUBTEST.
004332  042777  000004  174764  1$: BIC     #BIT2,@ASR      ;/TRY CLEARING BIT 2.
004340  005037  001124              CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
004344  017737  174754  001126      MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
004352  001401                      BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004354  104002                      ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                            ;/BIT 2 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004356                          2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 26        *TEST THAT CLOCK A STATUS REGISTER BIT 1 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILUPE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST.: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
004356  000004                  TST26:  SCOPE
004360  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
004366  005077  174732                  CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
004372  052777  000002  174724          BIS     #BIT1,@ASR      ;/SET BIT 1.
004400  012737  000002  001124          MOV     #BIT1,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004406  017737  174712  001126          MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
004414  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
004422  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004424  104002                        ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004426  000412                        BR      2$              ;/BR TC END SUBTEST.
004430  042777  000002  174666  1$:   BIC     #BIT1,@ASR      ;/TRY CLEARING BIT 1.
004436  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
004442  017737  174656  001126        MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
004450  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004452  104002                        ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

004454                          2$:
```

```
2310                                            ;/*
2311                                            ;;**********************************************************************
2312                                            ;*TEST 27       *TEST THAT CLOCK A STATUS REGISTER BIT 0 CAN BE SET AND CLEARED
2313                                            ;*
2314                                            ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
2315                                            ;*F/FS OR GATES
2316                                            ;*
2317                                            ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
2318                                            ;*
2319                                            ;*
2320                                            ;;**********************************************************************
2321  004454  000004                            TST27:  SCOPE
2322  004456  012737  000100  001164                    MOV     #100,$TIMES     ;;DO 100 ITERATIONS
2323  004464  005077  174634                            CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
2324  004470  052777  000001  174626                    BIS     #BIT0,@ASR      ;/SET BIT 0.
2325  004476  012737  000001  001124                    MOV     #BIT0,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
2326  004504  017737  174614  001126                    MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
2327  004512  023737  001124  001126                    CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
2328  004520  001402                                    BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
2329
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

2333  004522  104002                                    ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
2334                                                                            ;/BIT 0 FAILED TO BIT SET.
2335
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

2339  004524  000412                                    BR      2$
2340  004526  042777  000001  174570  1$:     BIC     #BIT0,@ASR      ;/TRY CLEARING BIT 0.
2341  004534  005037  001124                    CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
2342  004540  017737  174560  001126            MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
2343  004546  001401                            BEQ     2$              ;/IF ZERO-NO ERROR!
2344
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

2348  004550  104002                                    ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
2349                                                                            ;/BIT 0 FAILED TO CLEAR.
2350
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

2354  004552                            2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 30       *TEST THAT CLOCK A BUFFER REGISTER BIT 0 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
004552  000004                          TST30:  SCOPE
004554  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004562  005077  174540                          CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
004566  052777  000001  174532                  BIS     #BIT0,@ABR      ;/SET BIT 0.
004574  012737  000001  001124                  MOV     #BIT0,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004602  017737  174520  001126                  MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
004610  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
004616  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

004620  104003                                  ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 0 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

004622  000412                                  BR      2$              ;/BR TO END SUBTEST.
004624  042777  000001  174474  1$:             BIC     #BIT0,@ABR      ;/TRY CLEARING BIT 0.
004632  005037  001124                          CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
004636  017737  174464  001126                  MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
004644  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

004646  104003                                  ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 0 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

004650                                  2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 31        *TEST THAT CLOCK A BUFFER REGISTER BIT 1 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
2411  004650  000004           TST31:  SCOPE
2412  004652  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
2413  004660  005077  174442                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
2414  004664  052777  000002  174434          BIS     #BIT1,@ABR      ;/SET BIT 1.
2415  004672  012737  000002  001124          MOV     #BIT1,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
2416  004700  017737  174422  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
2417  004706  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
2418  004714  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
2419
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

2423  004716  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
2424                                                                  ;/BIT 1 FAILED TO BIT SET.
2425
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

2429  004720  000412                          BR      2$              ;/BR TO END SUBTEST.
2430  004722  042777  000002  174376  1$:     BIC     #BIT1,@ABR      ;/TRY CLEARING BIT 1.
2431  004730  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
2432  004734  017737  174366  001126          MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
2433  004742  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!
2434
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

2438  004744  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
2439                                                                  ;/BIT 1 FAILED TO CLEAR.
2440
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

2444  004746                    2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 32        *TEST THAT CLOCK A BUFFER REGISTER BIT 2 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
004746  000004                  TST32:  SCOPE
004750  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004756  005077  174344                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
004762  052777  000004  174336          BIS     #BIT2,@ABR      ;/SET BIT 2.
004770  012737  000004  001124          MOV     #BIT2,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
004776  017737  174324  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005004  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
005012  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005014  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 2 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005016  000412                          BR      2$              ;/BR TO END SUBTEST.
005020  042777  000004  174300  1$:     BIC     #BIT2,@ABR      ;/TRY CLEARING BIT 2.
005026  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
005032  017737  174270  001126          MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
005040  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005042  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 2 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005044                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 33       *TEST THAT CLOCK A BUFFER REGISTER BIT 3 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005044  000004                  TST33:  SCOPE
005046  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005054  005077  174246                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005060  052777  000010  174240          BIS     #BIT3,@ABR      ;/SET BIT 3.
005066  012737  000010  001124          MOV     #BIT3,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005074  017737  174226  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005102  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
005110  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005112  104003                        ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005114  000412                        BR      2$              ;/BR TO END SUBTEST.
005116  042777  000010  174202  1$:   BIC     #BIT3,@ABR      ;/TRY CLEARING BIT 3.
005124  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
005130  017737  174172  001126        MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
005136  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005140  104003                        ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

005142                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 34       *TEST THAT CLOCK A BUFFER REGISTER BIT 4 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005142  000004                        TST34:  SCOPE
005144  012737  000100  001164                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005152  005077  174150                        CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005156  052777  000020  174142                BIS     #BIT4,@ABR      ;/SET BIT 4.
005164  012737  000020  001124                MOV     #BIT4,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005172  017737  174130  001126                MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005200  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 4 AND ONLY BIT 4 SET?
005206  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005210  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 4 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005212  000412                                BR      2$              ;/BR TO END SUBTEST.
005214  042777  000020  174104  1$:           BIC     #BIT4,@ABR      ;/TRY CLEARING BIT 4.
005222  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
005226  017737  174074  001126                MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
005234  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005236  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 4 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005240                                  2$:
```

```
                                        ;/*
                                ;;*************************************************************
                                ;*TEST 35        *TEST THAT CLOCK A BUFFER REGISTER BIT 5 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;*************************************************************
005240  000004                  TST35:  SCOPE
005242  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005250  005077  174052                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005254  052777  000040  174044          BIS     #BIT5,@ABR      ;/SET BIT 5.
005262  012737  000040  001124          MOV     #BIT5,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005270  017737  174032  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005276  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
005304  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 36        *TEST THAT CLOCK A BUFFER REGISTER BIT 6 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005336  000004                        TST36:  SCOPE
005340  012737  000100  001164                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005346  005077  173754                        CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005352  052777  000100  173746                BIS     #BIT6,@ABR      ;/SET BIT 6.
005360  012737  000100  001124                MOV     #BIT6,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005366  017737  173734  001126                MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005374  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
005402  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005404  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 6 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005406  000412                                BR      2$              ;/BR TO END SUBTEST.
005410  042777  000100  173710  1$:           BIC     #BIT6,@ABR      ;/TRY CLEARING BIT 6.
005416  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
005422  017737  173700  001126                MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
005430  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005432  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 6 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005434                                  2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 37        *TEST THAT CLOCK A BUFFER REGISTER BIT 7 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005434  000004                  TST37:  SCOPE
005436  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005444  005077  173656                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005450  052777  000200  173650          BIS     #BIT7,@ABR      ;/SET BIT 7.
005456  012737  000200  001124          MOV     #BIT7,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005464  017737  173636  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005472  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
005500  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005502  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 7 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005504  000412                          BR      2$              ;/BR TO END SUBTEST.
005506  042777  000200  173612  1$:     BIC     #BIT7,@ABR      ;/TRY CLEARING BIT 7.
005514  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
005520  017737  173602  001126          MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
005526  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005530  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 7 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

005532                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 40       *TEST THAT CLOCK A BUFFER REGISTER BIT 8 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005532  000004                        TST40:  SCOPE
005534  012737  000100  001164                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005542  005077  173560                        CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005546  052777  000400  173552                BIS     #BIT8,@ABR      ;/SET BIT 8.
005554  012737  000400  001124                MOV     #BIT8,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005562  017737  173540  001126                MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005570  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 8 AND ONLY BIT 8 SET?
005576  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

005600  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 8 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

005602  000412                                BR      2$              ;/BR TO END SUBTEST.
005604  042777  000400  173514  1$:           BIC     #BIT8,@ABR      ;/TRY CLEARING BIT 8.
005612  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
005616  017737  173504  001126                MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
005624  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

005626  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 8 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

005630                                  2$:
```

```
                                        ;/*
                                        ;;*********************************************************************
                                        ;*TEST 41        *TEST THAT CLOCK A BUFFER REGISTER BIT 9 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*********************************************************************
005630  000004                  TST41:  SCOPE
005632  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005640  005077  173462                  CLR     @#BR            ;/CLEAR THE BUFFER REGISTER.
005644  052777  001000  173454          BIS     #BIT9,@#BR      ;/SET BIT 9.
005652  012737  001000  001124          MOV     #BIT9,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
005660  017737  173442  001126          MOV     @#BR,$BDDAT      ;/READ THE BUFFER REGISTER.
005666  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 9 AND ONLY BIT 9 SET?
005674  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005676  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 9 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005700  000412                          BR      2$              ;/BR TO END SUBTEST.
005702  042777  001000  173416  1$:     BIC     #BIT9,@#BR      ;/TRY CLEARING BIT 9.
005710  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
005714  017737  173406  001126          MOV     @#BR,$BDDAT     ;/NOW READ IT BACK.
005722  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005724  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 9 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

005726                          2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 42        *TEST THAT CLOCK A BUFFER REGISTER BIT 10 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
005726  000004                          TST42:  SCOPE
005730  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005736  005077  173364                          CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
005742  052777  002000  173356                  BIS     #BIT10,@ABR     ;/SET BIT 10.
005750  012737  002000  001124                  MOV     #BIT10,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
005756  017737  173344  001126                  MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
005764  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 10 AND ONLY BIT 10 SET?
005772  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

005774  104003                                  ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 10 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

005776  000412                                  BR      2$              ;/BR TO END SUBTEST.
006000  042777  002000  173320  1$:             BIC     #BIT10,@ABR     ;/TRY CLEARING BIT 10.
006006  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006012  017737  173310  001126                  MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
006020  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006022  104003                                  ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 10 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006024                                  2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 43        *TEST THAT CLOCK A BUFFER REGISTER BIT 11 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
006024  000004                          TST43:  SCOPE
006026  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006034  005077  173266                          CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
006040  052777  004000  173260                  BIS     #BIT11,@ABR     ;/SET BIT 11.
006046  012737  004000  001124                  MOV     #BIT11,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
006054  017737  173246  001126                  MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
006062  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 11 AND ONLY BIT 11 SET?
006070  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006072  104003                                  ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 11 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006074  000412                                  BR      2$              ;/BR TO END SUBTEST.
006076  042777  004000  173222  1$:             BIC     #BIT11,@ABR     ;/TRY CLEARING BIT 11.
006104  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006110  017737  173212  001126                  MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
006116  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006120  104003                                  ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 11 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006122                                  2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 44        *TEST THAT CLOCK A BUFFER REGISTER BIT 12 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
006122  000004                        TST44:  SCOPE
006124  012737  000100  001164                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006132  005077  173170                        CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
006136  052777  010000  173162                BIS     #BIT12,@ABR     ;/SET BIT 12.
006144  012737  010000  001124                MOV     #BIT12,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
006152  017737  173150  001126                MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
006160  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 12 AND ONLY BIT 12 SET?
006166  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

006170  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 12 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

006172  000412                                BR      2$              ;/BR TO END SUBTEST.
006174  042777  010000  173124  1$:           BIC     #BIT12,@ABR     ;/TRY CLEARING BIT 12.
006202  005037  001124                        CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006206  017737  173114  001126                MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
006214  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

006216  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 12 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

006220                                  2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 45        *TEST THAT CLOCK A BUFFER REGISTER BIT 13 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
006220  000004                          TST45:  SCOPE
006222  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006230  005077  173072                          CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
006234  052777  020000  173064                  BIS     #BIT13,@ABR     ;/SET BIT 13.
006242  012737  020000  001124                  MOV     #BIT13,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
006250  017737  173052  001126                  MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
006256  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 13 AND ONLY BIT 13 SET?
006264  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

006266  104003                                  ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 13 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

006270  000412                                  BR      2$              ;/BR TO END SUBTEST.
006272  042777  020000  173026  1$:             BIC     #BIT13,@ABR     ;/TRY CLEARING BIT 13.
006300  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006304  017737  173016  001126                  MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
006312  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

006314  104003                                  ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 13 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

006316                                  2$:
```

```
                                        ;/*
                                        ;;*********************************************************************
                                        ;*TEST 46       *TEST THAT CLOCK A BUFFER REGISTER BIT 14 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*********************************************************************
006316  000004                  TST46:  SCOPE
006320  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006326  005077  172774                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
006332  052777  040000  172766          BIS     #BIT14,@ABR     ;/SET BIT 14.
006340  012737  040000  001124          MOV     #BIT14,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
006346  017737  172754  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
006354  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 14 AND ONLY BIT 14 SET?
006362  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

006364  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 14 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

006366  000412                          BR      2$              ;/BR TO END SUBTEST.
006370  042777  040000  172730  1$:     BIC     #BIT14,@ABR     ;/TRY CLEARING BIT 14.
006376  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006402  017737  172720  001126          MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
006410  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

006412  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 14 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSSS

006414                          2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 47        *TEST THAT CLOCK A BUFFER REGISTER BIT 15 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK A BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
006414  000004                TST47:  SCOPE
006416  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006424  005077  172676                  CLR     @ABR            ;/CLEAR THE BUFFER REGISTER.
006430  052777  100000  172670          BIS     #BIT15,@ABR     ;/SET BIT 15.
006436  012737  100000  001124          MOV     #BIT15,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
006444  017737  172656  001126          MOV     @ABR,$BDDAT      ;/READ THE BUFFER REGISTER.
006452  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 15 AND ONLY BIT 15 SET?
006460  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

006462  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 15 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

006464  000412                          BR      2$              ;/BR TO END SUBTEST.
006466  042777  100000  172632  1$:     BIC     #BIT15,@ABR     ;/TRY CLEARING BIT 15.
006474  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006500  017737  172622  001126          MOV     @ABR,$BDDAT     ;/NOW READ IT BACK.
006506  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

006510  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 15 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

006512                          2$:
```

```
                                        ;/*
                                        ;;*********************************************************************
                                        ;*TEST 50        *TEST THAT CLOCK B STATUS REGISTER BIT 11 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*********************************************************************
006512  000004                  TST50:  SCOPE
006514  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006522  005077  172604                  CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
006526  052777  004000  172576          BIS     #BIT11,@BSR     ;/SET BIT 11.
006534  012737  004000  001124          MOV     #BIT11,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
006542  017737  172564  001126          MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
006550  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 11 AND ONLY BIT 11 SET?
006556  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSS

006560  104005                        ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 11 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSS

006562  000412                        BR      2$              ;/BR TO END SUBTEST.
006564  042777  004000  172540  1$:   BIC     #BIT11,@BSR     ;/TRY CLEARING BIT 11.
006572  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006576  017737  172530  001126        MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
006604  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSS

006606  104005                        ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 11 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS›› ERROR ‹‹SSSSSSSSSSSSSSSSSSSSSSS

006610                          2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 51        *TEST THAT CLOCK B STATUS REGISTER BIT 7 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
006610  000004                          TST51:  SCOPE
006612  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006620  005077  172506                          CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
006624  052777  000200  172500                  BIS     #BIT7,@BSR      ;/SET BIT 7.
006632  012737  000200  001124                  MOV     #BIT7,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
006640  017737  172466  001126                  MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
006646  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
006654  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006656  104005                                  ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 7 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006660  000412                                  BR      2$              ;/BR TO END SUBTEST.
006662  042777  000200  172442  1$:             BIC     #BIT7,@BSR      ;/TRY CLEARING BIT 7.
006670  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006674  017737  172432  001126                  MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
006702  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006704  104005                                  ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 7 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

006706                                  2$:
```

```
                                        ;/8
                                ;;**************************************************************
                                ;*TEST 52       *TEST THAT CLOCK B STATUS REGISTER BIT 6 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**************************************************************
006706  000004                  TST52:  SCOPE
006710  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006716  005077  172410                  CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
006722  052777  000100  172402          BIS     #BIT6,@BSR      ;/SET BIT 6.
006730  012737  000100  001124          MOV     #BIT6,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
006736  017737  172370  001126          MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
006744  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
006752  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

006754  104005                          ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 6 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

006756  000412                          BR      2$              ;/BR TO END SUBTEST.
006760  042777  000100  172344  1$:     BIC     #BIT6,@BSR      ;/TRY CLEARING BIT 6.
006766  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
006772  017737  172334  001126          MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007000  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007002  104005                          ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 6 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007004                          2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 53        *TEST THAT CLOCK B STATUS REGISTER BIT 5 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007004  000004                          TST53:  SCOPE
007006  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007014  005077  172312                          CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
007020  052777  000040  172304                  BIS     #BIT5,@BSR      ;/SET BIT 5.
007026  012737  000040  001124                  MOV     #BIT5,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007034  017737  172272  001126                  MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
007042  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
007050  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007052  104005                                  ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 5 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007054  000412                                  BR      2$              ;/BR TO END SUBTEST.
007056  042777  000040  172246  1$:             BIC     #BIT5,@BSR      ;/TRY CLEARING BIT 5.
007064  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007070  017737  172236  001126                  MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007076  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007100  104005                                  ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 5 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007102                                  2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 54        *TEST THAT CLOCK B STATUS REGISTER BIT 4 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
007102  000004                        TST54:  SCOPE
007104  012737  000100  001164                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007112  005077  172214                        CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
007116  052777  000020  172206                BIS     #BIT4,@BSR      ;/SET BIT 4.
007124  012737  000020  001124                MOV     #BIT4,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007132  017737  172174  001126                MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
007140  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 4 AND ONLY BIT 4 SET?
007146  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007150  104005                                ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 4 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007152  000412                                BR      2$              ;/BR TO END SUBTEST.
007154  042777  000020  172150  1$:           BIC     #BIT4,@BSR      ;/TRY CLEARING BIT 4.
007162  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
007166  017737  172140  001126                MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007174  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007176  104005                                ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 4 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007200                                  2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 55        *TEST THAT CLOCK B STATUS REGISTER BIT 3 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007200  000004                          TST55:  SCOPE
007202  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007210  005077  172116                          CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
007214  052777  000010  172110                  BIS     #BIT3,@BSR      ;/SET BIT 3.
007222  012737  000010  001124                  MOV     #BIT3,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007230  017737  172076  001126                  MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
007236  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
007244  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007246  104005                                  ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 3 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007250  000412                                  BR      2$              ;/BR TO END SUBTEST.
007252  042777  000010  172052  1$:             BIC     #BIT3,@BSR      ;/TRY CLEARING BIT 3.
007260  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007264  017737  172042  001126                  MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007272  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007274  104005                                  ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 3 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007276                                  2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 56        *TEST THAT CLOCK B STATUS REGISTER BIT 2 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007276  000004                          TST56:  SCOPE
007300  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007306  005077  172020                          CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
007312  052777  000004  172012                  BIS     #BIT2,@BSR      ;/SET BIT 2.
007320  012737  000004  001124                  MOV     #BIT2,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007326  017737  172000  001126                  MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
007334  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
007342  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007344  104005                                  ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 2 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007346  000412                                  BR      2$              ;/BR TO END SUBTEST.
007350  042777  000004  171754  1$:             BIC     #BIT2,@BSR      ;/TRY CLEARING BIT 2.
007355  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007362  017737  171744  001126                  MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007370  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007372  104005                                  ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 2 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007374                                  2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 57        *TEST THAT CLOCK B STATUS REGISTER BIT 1 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007374  000004                          TST57:  SCOPE
007376  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007404  005077  171722                          CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
007410  052777  000002  171714                  BIS     #BIT1,@BSR      ;/SET BIT 1.
007416  012737  000002  001124                  MOV     #BIT1,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007424  017737  171702  001126                  MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
007432  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
007440  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007442  104005                                  ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 1 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007444  000412                                  BR      2$              ;/BR TO END SUBTEST.
007446  042777  000002  171656  1$:             BIC     #BIT1,@BSR      ;/TRY CLEARING BIT 1.
007454  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007460  017737  171646  001126                  MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007466  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007470  104005                                  ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 1 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007472                                  2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 60       *TEST THAT CLOCK B STATUS REGISTER BIT 0 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;*****************************************************************
007472  000004                  TST60:  SCOPE
007474  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007502  005077  171624                  CLR     @BSR            ;/CLEAR THE STATUS REGISTER.
007506  052777  000001  171616          BIS     #BIT0,@BSR      ;/SET BIT 0.
007514  012737  000001  001124          MOV     #BIT0,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007522  017737  171604  001126          MOV     @BSR,$BDDAT      ;/READ THE STATUS REGISTER.
007530  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
007536  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007540  104005                        ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 0 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007542  000412                        BR      2$              ;/BR TO END SUBTEST.
007544  042777  000001  171560  1$:   BIC     #BIT0,@BSR      ;/TRY CLEARING BIT 0.
007552  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007556  017737  171550  001126        MOV     @BSR,$BDDAT     ;/NOW READ IT BACK.
007564  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007566  104005                        ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 0 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007570                          2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 61        *TEST THAT CLOCK B BUFFER REGISTER BIT 0 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007570  000004                          TST61:  SCOPE
007572  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007600  005077  171530                          CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
007604  052777  000001  171522                  BIS     #BIT0,@BBR      ;/SET BIT 0.
007612  012737  000001  001124                  MOV     #BIT0,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007620  017737  171510  001126                  MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
007626  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
007634  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007636  104006                                  ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                        ;/BIT 0 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007640  000412                                  BR      2$              ;/BR TO END SUBTEST.
007642  042777  000001  171464  1$:             BIC     #BIT0,@BBR      ;/TRY CLEARING BIT 0.
007650  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007654  017737  171454  001126                  MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
007662  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007664  104006                                  ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                        ;/BIT 0 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

007666                                  2$:
```

```
                                        ;/*
                                ;;**************************************************************
                                ;*TEST 62        *TEST THAT CLOCK B BUFFER REGISTER BIT 1 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**************************************************************
007666  000004                  TST62:  SCOPE
007670  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007676  005077  171432                  CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
007702  052777  000002  171424          BIS     #BIT1,@BBR      ;/SET BIT 1.
007710  012737  000002  001124          MOV     #BIT1,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
007716  017737  171412  001126          MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
007724  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
007732  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007734  104006                          ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 1 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007736  000412                          BR      2$              ;/BR TO END SUBTEST.
007740  042777  000002  171366  1$:     BIC     #BIT1,@BBR      ;/TRY CLEARING BIT 1.
007746  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
007752  017737  171356  001126          MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
007760  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007762  104006                          ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 1 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

007764                          2$:
```

```
                                                ;/#
                                        ;;*****************************************************************
                                        ;*TEST 63        *TEST THAT CLOCK B BUFFER REGISTER BIT 2 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007764  000004                          TST63:  SCOPE
007766  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007774  005077  171334                          CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
010000  052777  000004  171326                  BIS     #BIT2,@BBR      ;/SET BIT 2.
010006  012737  000004  001124                  MOV     #BIT2,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
010014  017737  171314  001126                  MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
010022  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
010030  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

010032  104006                                  ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                        ;/BIT 2 FAILED TO BIT SET.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

010034  000412                                  BR      2$              ;/BR TO END SUBTEST.
010036  042777  000004  171270  1$:             BIC     #BIT2,@BBR      ;/TRY CLEARING BIT 2.
010044  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
010050  017737  171260  001126                  MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
010056  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

010060  104006                                  ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                        ;/BIT 2 FAILED TO CLEAR.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

010062                                  2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 64        *TEST THAT CLOCK B BUFFER REGISTER BIT 3 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
010062  000004                  TST64:  SCOPE
010064  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
010072  005077  171236                  CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
010076  052777  000010  171230          BIS     #BIT3,@BBR      ;/SET BIT 3.
010104  012737  000010  001124          MOV     #BIT3,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
010112  017737  171216  001126          MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
010120  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
010126  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010130  104006                          ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010132  000412                          BR      2$              ;/BR TO END SUBTEST.
010134  042777  000010  171172  1$:     BIC     #BIT3,@BBR      ;/TRY CLEARING BIT 3.
010142  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
010146  017737  171162  001126          MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
010154  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010156  104006                          ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010160                          2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 65        *TEST THAT CLOCK B BUFFER REGISTER BIT 4 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*
                                ;;**********************************************************************
010160  000004                TST65:  SCOPE
010162  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
010170  005077  171140                CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
010174  052777  000020  171132        BIS     #BIT4,@BBR      ;/SET BIT 4.
010202  012737  000020  001124        MOV     #BIT4,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
010210  017737  171120  001126        MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
010216  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DID BIT 4 AND ONLY BIT 4 SET?
010224  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010226  104006                        ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 4 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010230  000412                        BR      2$              ;/BR TO END SUBTEST.
010232  042777  000020  171074  1$:   BIC     #BIT4,@BBR      ;/TRY CLEARING BIT 4.
010240  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
010244  017737  171064  001126        MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
010252  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010254  104006                        ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 4 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010256                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 66        *TEST THAT CLOCK B BUFFER REGISTER BIT 5 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
010256  000004                          TST66:  SCOPE
010260  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
010266  005077  171042                          CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
010272  052777  000040  171034                  BIS     #BIT5,@BBR      ;/SET BIT 5.
010300  012737  000040  001124                  MOV     #BIT5,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
010306  017737  171022  001126                  MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
010314  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
010322  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010324  104006                                ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                        ;/BIT 5 FAILED TO BIT SET.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010326  000412                                BR      2$              ;/BR TO END SUBTEST.
010330  042777  000040  170776  1$:           BIC     #BIT5,@BBR      ;/TRY CLEARING BIT 5.
010336  005037  001124                        CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
010342  017737  170766  001126                MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
010350  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010352  104006                                ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                        ;/BIT 5 FAILED TO CLEAR.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010354                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 67        *TEST THAT CLOCK B BUFFER REGISTER BIT 6 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*
                                        ;;**********************************************************************
010354  000004                        TST67:  SCOPE
010356  012737  000100  001164                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
010364  005077  170744                        CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
010370  052777  000100  170736                BIS     #BIT6,@BBR      ;/SET BIT 6.
010376  012737  000100  001124                MOV     #BIT6,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
010404  017737  170724  001126                MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
010412  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
010420  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010422  104006                              ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                    ;/BIT 6 FAILED TO BIT SET.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010424  000412                          1$:     BR      2$              ;/BR TO END SUBTEST.
010426  042777  000100  170700                BIC     #BIT6,@BBR      ;/TRY CLEARING BIT 6.
010434  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
010440  017737  170670  001126                MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
010446  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010450  104006                              ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                    ;/BIT 6 FAILED TO CLEAR.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

010452                          2$:
```

```
3805                                            ;/*
3806                                            ;;**********************************************************************
3807                                            ;*TEST 70        *TEST THAT CLOCK B BUFFER REGISTER BIT 7 CAN BE SET AND CLEARED
3808                                            ;*
3809                                            ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
3810                                            ;*F/FS OR GATES
3811                                            ;*
3812                                            ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
3813                                            ;*
3814                                            ;*
3815                                            ;;**********************************************************************
3816  010452  000004                            TST70:  SCOPE
3817  010454  012737  000100  001164                    MOV     #100,$TIMES     ;;DO 100 ITERATIONS
3818  010462  005077  170646                            CLR     @BBR            ;/CLEAR THE BUFFER REGISTER.
3819  010466  052777  000200  170640                    BIS     #BIT7,@BBR      ;/SET BIT 7.
3820  010474  012737  000200  001124                    MOV     #BIT7,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
3821  010502  017737  170626  001126                    MOV     @BBR,$BDDAT      ;/READ THE BUFFER REGISTER.
3822  010510  023737  001124  001126                    CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
3823  010516  001402                                    BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
3824
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

3828  010520  104006                                    ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
3829                                                                            ;/BIT 7 FAILED TO BIT SET.
3830
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

3834  010522  000412                                    BR      2$              ;/BR TO END SUBTEST.
3835  010524  042777  000200  170602  1$:             BIC     #BIT7,@BBR      ;/TRY CLEARING BIT 7.
3836  010532  005037  001124                          CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
3837  010536  017737  170572  001126                  MOV     @BBR,$BDDAT     ;/NOW READ IT BACK.
3838  010544  001401                                    BEQ     2$              ;/IF ZERO-NO ERROR!
3839
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

3843  010546  104006                                    ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
3844                                                                            ;/BIT 7 FAILED TO CLEAR.
3845
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

3849  010550                                    2$:
3850                                            .SBTTL  *
3851                                            .SBTTL  * PHASE 2 ADVANCED BASIC LOGIC TESTS
3852                                            .SBTTL  *
3853
3854
3855
3856                                            ;;**********************************************************************
3857                                            ;*TEST 71        *TEST THE LOW BYTE OPERATION OF CLOCK A'S STATUS REGISTER
3858
3859                                            ;*
3860                                            ;*WE CAN SUCCESSFULLY WRITE EVERY BIT IN STATUS REG A
```

```
                                        ;*NOW LETS CHECK THE BYTE OPERATION OF THIS REGISTER.
                                        ;*
                                        ;*FOR   LOW BYTE        CO=H; C1=L; ADD=H GIVEING "LD STAT A LO" H
                                        ;*                      AND NOT "LD STAT A HI" H
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
                                        ;*
                                        ;*FOR   HIGH BYTE       CO=H; C1=L; ADD=L GIVING "LD STAT A HI" H
                                        ;*                      AND NOT "LD STAT LO" H
                                        ;*
                                        ;**********************************  ******************************
010550  000004                          TST71:  SCOPE
010552  012737  000050  001164                  MOV     #50,$TIMES      ;;DO 50 ITERATIONS

010560  005077  170540                          CLR     @ASR            ;MAKE SURE THE STATUS REGISTER IS CLEAR.
010564  112777  127677  170532                  MOVB    #127677,@ASR    ;TRY WRITING ALL BITS IN THE
                                                                        ;STATUS REGISTER. LOGIC SHOULD PREVENT IT
                                                                        ;FROM BEING WRITTEN INTO BECAUSE
                                                                        ;WE ARE USING A DATOB INSTRUCTION.

010572  117777  170526  170326                  MOVB    @ASR,@$BDDAT    ;NOW EXAMINE THE
                                                                        ;STATUS REGISTER.
010600  013737  001126  001124                  MOV     $BDDAT,$GDDAT   ;FIX $GDDAT FOR ERROR TYPEOUT IF
010606  105037  001125                          CLRB    $GDDAT+1        ;ANY ERROR HAS OCCURRED, UPPER BYTE CLEARED.

010612  105737  001127                          TSTB    $BDDAT+1        ;ARE ANY BITS IN THE UPPER BYTE
                                                                        ;OF THE STATUS REGISTER SET?
010616  001401                                  BEQ     1$              ;BRANCH NEXT TEST IF UPPER BYTE=0.

        ;;;$$$
```

```
3917                                    ;*
3918                                    ;*FOR   HIGH BYTE       CO=H; C1=L; ADD=L GIVING "LD STAT A HI" H
3919                                    ;*                      AND NOT "LD STAT LO" H
3920                                    ;*
3921                                    ;;*****************************************************************
3922  010622  000004                    TST72:  SCOPE
3923  010624  012737  000050  001164            MOV     #50,$TIMES      ;;DO 50 ITERATIONS
3924
3925  010632  005077  170466                    CLR     @ASR            ;CLEAR THE STATUS REGISTER.
3926
3927  010636  005237  001324                    INC     ASR             ;ADD #1 TO THE STATUS REGISTER'S ADDRESS
3928                                                                    ;SO THAT WE WILL BE WRITING INTO
3929                                                                    ;THE HIGH BYTE.
3930
3931  010642  112777  177313  170454          MOVB    #177313,@ASR    ;TRY WRITING ALL BITS IN THE STATUS
3932                                                                    ;REGISTER. LOGIC SHOULD PREVENT THE LOW
3933                                                                    ;BYTE OF THE STATUS REGISTER FROM
3934                                                                    ;BEING WRITTEN INTO BECAUSE WE ARE USING
3935                                                                    ;A DATOB INSTRUCTION WITH ADD SET.
3936
3937  010650  005337  001324                    DEC     ASR             ;FIX ADDRESS OF THE STATUS REGISTER ADDR.
3938                                                                    ;SO WE CAN LOOK AT THE WHOLE WORD.
3939  010654  017737  170444  001126          MOV     @ASR,$BDDAT     ;READ BACK WHAT THE STATUS REG. CONTAINS
3940  010662  013737  001126  001124          MOV     $BDDAT,$GDDAT   ;FIX $GDDAT FOR ERROR TYPEOUT IF AN ERROR
3941  010670  105037  001124                  CLRB    $GDDAT          ;OCCURRED, LOWER BYTE CLEARED.
3942  010674  105737  001126                  TSTB    $BDDAT          ;IS LOWER BYTE CLEAR?
3943  010700  001401                          BEQ     1$              ;BR IF YES TO NEXT SUBTEST.
3944
3945

        ;;;SSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSS

3949  010702  104001                          ERROR   1               ;ERROR - WROTE INTO LOWER BYTE
3950                                                                  ;OF CLOCK A'S STATUS REGISTER WHEN
3951                                                                  ;DOING A DATOB TO THE HIGH BYTE.
3952
3953

        ;;;SSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSS

3957  010704                    1$:
3958
3959
3960
3961                                    ;;*****************************************************************
3962                                    ;*TEST 73        *TEST THE LOW BYTE OPERATION OF B'S STATUS REGISTER
3963
3964                                    ;*
3965                                    ;*WE CAN SUCCESSFULLY WRITE EVERY BIT IN STATUS REG A
3966                                    ;*NOW LETS CHECK THE BYTE OPERATION OF THIS REGISTER.
3967                                    ;*
3968                                    ;*FOR   LOW BYTE         CO=H; C1=L; ADD=H GIVEING "LD STAT A LO" H
3969                                    ;*                      AND NOT "LD STAT A HI" H
3970                                    ;*
3971                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
3972                                    ;*
```

```
                                        ;*
                                        ;*FOR   HIGH BYTE       CO=H; C1=L; ADD=L GIVING "LD STAT A HI" H
                                        ;*                      AND NOT "LD STAT LO" H
                                        ;*
                                        ;;*****************************************************************
010704  000004                  TST73:  SCOPE
010706  012737  000050  001164          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
010714  005077  170412                  CLR     @BSR            ;MAKE SURE THE STATUS REGISTER IS CLEAR
010720  112777  137677  170404          MOVB    #137677,@BSR    ;TRY WRITING ALL BITS IN THE
                                                                ;STATUS REGISTER. LOGIC SHOULD PREVENT
                                                                ;FROM BEING WRITTEN INTO BECAUSE
                                                                ;WE ARE USING A DATOB INSTRUCTION.
010726  017737  170400  001126          MOV     @BSR,$BDDAT     ;NOW EXAMINE THE
                                                                ;STATUS REGISTER
010734  013737  001126  001124          MOV     $BDDAT,$GDDAT   ;FIX $GDDAT FOR ERROR TYPEOUT IF
010742  105037  001125                  CLRB    $GDDAT+1        ;ANY ERROR HAS OCCURRED, UPPER BYTE CLEARED.
010746  105737  001127                  TSTB    $BDDAT+1        ;ARE ANY BITS IN THE UPPER BYTE
                                                                ;OF THE STATUS REGISTER SET?
010752  001401                          BEQ     1$              ;BRANCH NEXT TEST IF UPPER BYTE=0

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010754  104005                          ERROR   5               ;ERROR - WROTE INTO UPPER BYTE OF
                                                                ;CLOCK B'S STATUS WHEN
                                                                ;DOING A DATOB TO THE LOW BYTE.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

010756                          1$:


                                        ;;*****************************************************************
                                        ;*TEST 74        *TEST THE HIGH BYTE OPERATION OF B'S STATUS REGISTER
                                        ;*
                                        ;*WE CAN SUCCESSFULLY WRITE EVERY BIT IN STATUS REG B
                                        ;*NOW LETS CHECK THE BYTE OPERATION OF THIS REGISTER.
                                        ;*
                                        ;*FOR   LOW BYTE         CO=H; C1=L; ADD=H GIVEING "LD STAT B LO" H
                                        ;*                      AND NOT "LD STAT B HI" H
                                        ;*
                                        ;*FOR   HIGH BYTE        CO=H; C1=L; ADD=L GIVING "LD STAT B HI" H
                                        ;*                      AND NOT "LD STAT LO" H
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
                                        ;*
                                        ;*
                                        ;;*****************************************************************
010756  000004                  TST74:  SCOPE
```

```
4029  010760  012737  000050  001164          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
4030  010766  005077  170340                  CLR     @BSR            ;CLEAR THE STATUS REGISTER.
4031
4032  010772  005237  001332                  INC     BSR             ;ADD #1 TO THE STATUS REGISTER'S ADDRESS
4033                                                                    ;SO THAT WE WILL BE WRITING INTO
4034                                                                    ;THE HIGH BYTE.
4035
4036  010776  112777  177777  170326          MOVB    #177777,@BSR    ;TRY WRITING ALL BITS IN THE STATUS
4037                                                                    ;REGISTER. LOGIC SHOULD PREVENT THE LOW
4038                                                                    ;BYTE OF THE STATUS REGISTER FROM
4039                                                                    ;BEING WRITTEN INTO BECAUSE WE ARE USING
4040                                                                    ;A DATOB INSTRUCTION WITH ADD SET.
4041
4042  011004  005337  001332                  DEC     BSR             ;FIX ADDRESS OF THE STATUS REGISTER ADDR.
4043                                                                    ;SO WE CAN LOOK AT THE WHOLE WORD.
4044  011010  017737  170316  001126          MOV     @BSR,$BDDAT     ;READ BACK WHAT THE STATUS REG. CONTAINS
4045  011016  013737  001126  001124          MOV     $BDDAT,$GDDAT   ;FIX $GDDAT FOR ERROR TYPEOUT IF AN ERROR
4046  011024  105037  001124                  CLRB    $GDDAT          ;OCCURRED, LOWER BYTE CLEARED.
4047  011030  105737  001126                  TSTB    $BDDAT          ;IS LOWER BYTE CLEAR?
4048  011034  001401                          BEQ     1$              ;BR IF YES TO NEXT SUBTEST.
4049
4050

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4054  011036  104005                          ERROR   5               ;ERROR - WROTE INTO LOWER BYTE
4055                                                                    ;OF CLOCK B'S STATUS REGISTER WHEN
4056                                                                    ;DOING A DATOB TO THE HIGH BYTE.
4057
4058

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4062  011040                          1$:
4063
4064
4065                                    ;;********************************************************************
4066                                    ;*TEST 75        *TEST THAT CLOCK A'S COUNT REGISTER IS CLEAR
4067
4068                                    ;*
4069                                    ;*CLOCK A COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
4070                                    ;*F/FS OR GATES
4071                                    ;*
4072                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
4073                                    ;*
4074                                    ;*
4075
4076                                    ;;********************************************************************
4077  011040  000004                  TST75:  SCOPE
4078  011042  012737  000002  001164          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
4079
4080  011050  005077  170250                  CLR     @ASR            ;SELECT MODE 0.
4081  011054  005077  170246                  CLR     @ABR            ;CLEAR THE BUFFER REGISTER. BUFFER
4082                                                                    ;REGISTER WILL BE TRANSFERRED TO
4083                                                                    ;COUNT REGISTER SINCE THIS IS MODE 0.
4084
```

```
 4085  011060  005037  001124                    CLR     $GDDAT          ;EXPECT TO READ BACK ALL 0'S.
 4086  011064  017737  170240  001126            MOV     @ACR,$BDDAT     ;READ THE COUNT REGISTER - IT SHOULD BE CLEAR.
 4087  011072  001401                            BEQ     1$              ;BR IF YES TO NEXT TEST
 4088
 4089
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

 4093  011074  104004                            ERROR   4               ;ERROR - CLOCK A'S COUNTER REGISTER
 4094                                                                    ;NOT CLEAR.
 4095
 4096
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

 4100  011076                          1$:
 4101
 4102                                   ;;*********************************************************************
 4103                                   ;*TEST 76        *TEST CLOCK A'S COUNT REGISTER WITH 125252 PATTERN
 4104
 4105                                   ;*
 4106                                   ;*CLOCK A COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
 4107                                   ;*F/FS OR GATES
 4108                                   ;*
 4109                                   ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
 4110                                   ;*
 4111                                   ;*
 4112
 4113                                   ;;*********************************************************************
 4114  011076  000004                  TST76:  SCOPE
 4115  011100  012737  000100  001164           MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 4116
 4117  011106  005077  170212                    CLR     @ASR            ;SELECT MODE 0.
 4118  011112  012777  125252  170206            MOV     #125252,@ABR    ;LOAD THE BUFFER REGISTER WITH
 4119                                                                    ;PATTERN 125252. IT WILL BE
 4120                                                                    ;TRANSFERRED TO THE COUNT REGISTER
 4121                                                                    ;SINCE THIS IS MODE 0.
 4122
 4123  011120  012737  125252  001124            MOV     #125252,$GDDAT  ;SET EXPECTED TO PATTERN IN CASE OF
 4124                                                                    ;NEED OF ERROR TYPEOUT.
 4125  011126  017737  170176  001126            MOV     @ACR,$BDDAT     ;READ THE COUNT REGISTER
 4126
 4127  011134  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
 4128                                                                    ;COME THROUGH?
 4129  011142  001401                            BEQ     1$              ;BR IF YES TO NEXT TEST.
 4130
 4131
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

 4135  011144  104004                            ERROR   4               ;DATA ERROR CLOCK A + PATTERN "125252"
 4136                                                                    ;FAILED TO TRANSFER PROPERLY BETWEEN
 4137                                                                    ;BUFFER AND COUNT REGISTERS.
 4138
 4139
                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS
```

```
4143  011146                          1$:
4144
4145                                  ;;**********************************************************************
4146                                  ;*TEST 77        *TEST CLOCK A'S COUNT REGISTER WITH 052525 PATTERN
4147
4148                                  ;*
4149                                  ;*CLOCK A COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
4150                                  ;*F/FS OR GATES
4151                                  ;*
4152                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
4153                                  ;*
4154                                  ;*
4155
4156                                  ;;**********************************************************************
4157  011146  000004                  TST77:  SCOPE
4158  011150  012737  000050  001164          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
4159
4160  011156  005077  170142          CLR     @ASR            ;SELECT MODE 0.
4161  011162  012777  052525  170136  MOV     #052525,@ABR    ;LOAD THE BUFFER REGISTER WITH
4162                                                          ;PATTERN 052525. IT WILL BE
4163                                                          ;TRANSFERRED TO THE COUNT REGISTER
4164                                                          ;SINCE THIS IS MODE 0.
4165
4166  011170  012737  052525  001124  MOV     #052525,$GDDAT  ;SET EXPECTED TO PATTERN IN CASE OF
4167                                                          ;NEED OF ERROR TYPEOUT.
4168  011176  017737  170126  001126  MOV     @ACR,$BDDAT     ;READ THE COUNT REGISTER
4169
4170  011204  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
4171                                                          ;COME THROUGH?
4172  011212  001401                  BEQ     1$              ;BR IF YES TO NEXT TEST.
4173
4174
        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4178  011214  104004                  ERROR   4               ;DATA ERROR CLOCK A + PATTERN "052525"
4179                                                          ;FAILED TO TRANSFER PROPERLY BETWEEN
4180                                                          ;BUFFER AND COUNT REGISTERS.
4181
4182
        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4186  011216                  1$:
4187
4188
4189                                  ;;**********************************************************************
4190                                  ;*TEST 100       *TEST THAT CLOCK B'S COUNT REGISTER IS CLEAR
4191
4192                                  ;*
4193                                  ;*CLOCK B COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
4194                                  ;*F/FS OR GATES
4195                                  ;*
4196                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
```

```
                                        ;*
                                        ;*

                                        ;;*****************************************************************
011216  000004                          TST100: SCOPE
011220  012737  000002  001164                  MOV     #2,$TIMES       ;;DO 2 ITERATIONS

011226  005077  170100                          CLR     @BSR            ;SELECT MODE 0.
011232  005077  170076                          CLR     @BBR            ;CLEAR THE BUFFER REGISTER. BUFFER
                                                                        ;REGISTER WILL BE TRANSFERRED TO
                                                                        ;COUNT REGISTER SINCE THIS IS MODE 0.

011236  005037  001124                          CLR     $GDDAT          ;EXPECT TO READ BACK ALL 0'S.
011242  017737  170070  001126                  MOV     @BCR,$BDDAT     ;READ THE COUNT REGISTER - IT SHOULD BE CLEAR.
011250  001401                                  BEQ     1$              ;BR IF YES TO NEXT TEST


        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

011252  104007                                  ERROR   7               ;ERROR - CLOCK B'S COUNTER REGISTER
                                                                        ;NOT CLEAR.


        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

011254                                  1$:


                                        ;;*****************************************************************
                                        ;*TEST 101      *TEST CLOCK B'S COUNT REGISTER WITH 125 PATTERN

                                        ;*
                                        ;*CLOCK B COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                        ;*
                                        ;*

                                        ;;*****************************************************************
011254  000004                          TST101: SCOPE
011256  012737  000100  001164                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS

011264  005077  170042                          CLR     @BSR            ;SELECT MODE 0.
011270  012777  000125  170036                  MOV     #125,@BBR       ;LOAD THE BUFFER REGISTER WITH
                                                                        ;PATTERN 125. IT WILL BE
                                                                        ;TRANSFERRED TO THE COUNT REGISTER
                                                                        ;SINCE THIS IS MODE 0.

011276  012737  000125  001124                  MOV     #125,$GDDAT     ;SET EXPECTED TO PATTERN IN CASE OF
                                                                        ;NEED OF ERROR TYPEOUT.
011304  017737  170026  001126                  MOV     @BCR,$BDDAT     ;READ THE COUNT REGISTER

011312  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
```

```
                                                        ;COME THROUGH?
011320  001401                          BEQ     1$      ;BR IF YES TO NEXT TEST.


        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

011322  104007                          ERROR   7       ;DATA ERROR CLOCK B - PATTERN "125"
                                                        ;FAILED TO TRANSFER PROPERLY BETWEEN
                                                        ;BUFFER AND COUNT REGISTERS.


        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

011324                          1$:

                                ;;**************************************************************
                                ;*TEST 102       *TEST CLOCK B'S COUNT REGISTER WITH 252 PATTERN

                                ;*
                                ;*CLOCK B COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
                                ;*
                                ;*

                                ;;**************************************************************
011324  000004                  TST102: SCOPE
011326  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS

011334  005077  167772                  CLR     @BSR            ;SELECT MODE 0.
011340  012777  000252  167766          MOV     #252,@BBR       ;LOAD THE BUFFER REGISTER WITH
                                                                ;PATTERN 252. IT WILL BE
                                                                ;TRANSFERRED TO THE COUNT REGISTER
                                                                ;SINCE THIS IS MODE 0.

011346  012737  000252  001124          MOV     #252,$GDDAT     ;SET EXPECTED TO PATTERN IN CASE OF
                                                                ;NEED OF ERROR TYPEOUT.
011354  017737  167756  001126          MOV     @BCR,$BDDAT     ;READ THE COUNT REGISTER

011362  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
                                                                ;COME THROUGH?
011370  001401                          BEQ     1$              ;BR IF YES TO NEXT TEST.


        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

011372  104007                          ERROR   7       ;DATA ERROR CLOCK B - PATTERN "252"
                                                        ;FAILED TO TRANSFER PROPERLY BETWEEN
                                                        ;BUFFER AND COUNT REGISTERS.


        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS
```

```
MAINDEC-11-DZKWK-A     MACY11 27(732)  26-OCT-76  10:49  PAGE 99
DZKWK.CMB       T102   *TEST CLOCK B'S COUNT REGISTER WITH 252 PATTERN


 4311  011374                          1$:
 4312
 4313
 4314                                   ;;**********************************************************************
 4315                                   ;*TEST 103      *TEST THAT INIT CLEARS STATUS REGISTER A
 4316                                   ;*
 4317                                   ;*TESTING OF THE INIT LOGIC AS RECEIVED FROM THE UNIBUS AND BUFFERED
 4318                                   ;*TO STATUS REGISTER F/FS.
 4319                                   ;*
 4320                                   ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" - WILL BE ROUGH SYNCING
 4321                                   ;*
 4322                                   ;*
 4323                                   ;;**********************************************************************
 4324  011374  000004                  TST103: SCOPE
 4325  011376  012737  000005  001164          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
 4326
 4327  011404  005037  001124          CLR     $GDDAT          ;EXPECTED DATA IS ZERO.
 4328  011410  012777  177777  167706  MOV     #177777,@ASR    ;SET ALL BITS IN THE STATUS REG.
 4329
 4330  011416  000005                  RESET                   ;SYSTEM INITIALIZE.
 4331
 4332  011420  017737  167700  001126  MOV     @ASR,$BDDAT     ;READ THE STATUS REG., ALL BITS SHOULD
 4333                                                          ;HAVE BEEN CLEARED BY INIT.
 4334  011426  001401                  BEQ     1$              ;BR IF YES TO NEXT TEST.
 4335
 4336
          ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4340  011430  104002                  ERROR   2               ;ERROR - SYSTEM INIT FAILED TO CLEAR
 4341                                                          ;STATUS REGISTER CLOCK A.
 4342
 4343
          ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4347  011432                          1$:
 4348
 4349                                   ;;**********************************************************************
 4350                                   ;*TEST 104      *TEST THAT INIT CLEARS BUFFER REGISTER A
 4351                                   ;*
 4352                                   ;*THIS TEST IS DESIGNED TO SEE IF "INIT H"
 4353                                   ;*CLEARS THE BUFFER REGISTER. WE ALREADY
 4354                                   ;*KNOW IT CLEARS THE STATUS REG.
 4355                                   ;*
 4356                                   ;*
 4357                                   ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
 4358                                   ;*
 4359                                   ;;**********************************************************************
 4360  011432  000004                  TST104: SCOPE
 4361  011434  012737  000005  001164          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
 4362
 4363  011442  005037  001124          CLR     $GDDAT          ;CLEAR EXPECTED DATA.
 4364  011446  012777  177777  167652  MOV     #177777,@ABR    ;SET ALL BITS IN THE BUFFER REGISTER.
```

```
011454  000005                          RESET                   ;ISSUE SYSTEM INITIALIZE.

011456  017737  167644  001126          MOV     @ABR,$BDDAT     ;READ THE BUFFER REGISTER, ALL BITS
                                                                ;SHOULD HAVE BEEN CLEARED BY INIT.
011464  001401                          BEQ     1$              ;BR IF YES TO NEXT SUBTEST.


        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

011466  104003                          ERROR   3               ;ERROR - SYSTEM INIT FAILED
                                                                ;TO CLEAR BUFFER REGISTER A.


        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

011470                          1$:

                                ;;*****************************************************************
                                ;*TEST 105      *TEST THAT INIT CLEARS STATUS REGISTER B
                                ;*
                                ;*THIS TEST WILL FIND OUT IF "INIT H"
                                ;*CLEARS CLOCK B'S STATUS REG. WE
                                ;*ALREADY KNOW THAT IT CLEARS A'S STAT. REG.
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                ;*
                                ;;*****************************************************************
011470  000004                  TST105: SCOPE
011472  012737  000005  001164          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

011500  005037  001124                  CLR     $GDDAT          ;CLEAR EXPECTED DATA.
011504  012777  000377  167620          MOV     #377,@BSR       ;SET ALL BITS IN THE STATUS REGISTER.

011512  000005                          RESET                   ;ISSUE SYSTEM INITIALIZE.

011514  017737  167612  001126          MOV     @BSR,$BDDAT     ;READ THE STATUS REGISTER, ALL BITS
                                                                ;SHOULD HAVE BEEN CLEARED BY INIT.
011522  001401                          BEQ     1$              ;BR IF YES TO NEXT SUBTEST.


        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

011524  104005                          ERROR   5               ;ERROR - SYSTEM INIT FAILED
                                                                ;TO CLEAR STATUS REGISTER B.
```

```
 4414
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4418  011526                          1$:
```

```
                                ;;*****************************************************************
                                ;*TEST 106      *TEST THAT INIT CLEARS BUFFER REGISTER B
                                ;*
                                ;*NOW WE'LL SEE IF "INIT H" CLEARS
                                ;*CLOCK B'S BUFFER REGISTER. WE ALREADY
                                ;*NOW IT'LL CLEAR B'S STATUS REG.
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUF B"
                                ;*
                                ;;*****************************************************************
011526  000004                  TST106: SCOPE
011530  012737  000005  001164          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

011536  005037  001124          CLR     $GDDAT          ;CLEAR EXPECTED DATA.
011542  012777  000377  167564  MOV     #377,@BBR       ;SET ALL BITS IN THE BUFFER REGISTER.

011550  000005                  RESET                   ;ISSUE SYSTEM INITIALIZE.

011552  017737  167556  001126  MOV     @BBR,$BDDAT     ;READ THE BUFFER REGISTER, ALL BITS
                                                        ;SHOULD HAVE BEEN CLEARED BY INIT.
011560  001401                  BEQ     1$              ;BR IF YES TO NEXT SUBTEST.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

011562  104006                  ERROR   6               ;ERROR - SYSTEM INIT FAILED
                                                        ;TO CLEAR BUFFER REGISTER B.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

011564                          1$:


                                ;;*****************************************************************
                                ;*TEST 107      *TEST THE SETTING OF MAINTENANCE STP1 IN CLOCK A BIT 15 TO SET
                                ;*
                                ;*NEW SIGNALS   GENERATION OF "STP1" BY "LD STAT A" H + "BD 12" H
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
                                ;*
                                ;*
                                ;;*****************************************************************
011564  000004                  TST107: SCOPE

011566  005077  167532          CLR     @ASR            ;MAKE SURE THE STATUS REGISTER IS CLEAR.
011572  052777  010000  167524  BIS     #BIT12,@ASR     ;SET MAINTENANCE STP1.

011600  005777  167520          TST     @ASR            ;DID BIT15 (STP1 FLAG) SET?
011604  100401                  BMI     1$              ;BR IF YES - NEXT TEST
```

```
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4478  011606  104001                          ERROR   1               ;ERROR - MAINTENANCE STP1 (BIT12)
 4479                                                                  ;DID NOT SET BIT15 (STP1 FLAG) CLOCK A.
 4480
 4481
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4485  011610                          1$:
 4486
 4487
 4488                                   ;;*****************************************************************
 4489                                   ;*TEST 110      *TEST THAT BIT00 IN CLOCK A STATUS REG. WILL SET WHEN BIT13 AND MAIN. ST
 4490                                   ;*
 4491                                   ;*NEW SIGNALS   "STP1" H + "ST1 ENB CNTR (1)" H DIRECT SETTING
 4492                                   ;*             "ST1 FLAG"
 4493                                   ;*
 4494                                   ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
 4495                                   ;*
 4496                                   ;*
 4497                                   ;;*****************************************************************
 4498  011610  000004                  ST110: SCOPE
 4499
 4500  011612  012777  020000  167504          MOV     #BIT13,@ASR     ;SET "ST1 ENB COUNTER" IN CLK A'S STATUS REG.
 4501  011620  052777  010000  167476          BIS     #BIT12,@ASR     ;GENERATE A MAINTENANCE ST1.
 4502  011626  032777  000001  167470          BIT     #BIT00,@ASR     ;DID BIT00 (ENABL CNTR A) SET?
 4503  011634  001001                          BNE     1$              ;BR IF YES - NEXT TEST.
 4504
 4505
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4509  011636  104001                          ERROR   1               ;ERROR - BIT00 OF CLOCK A'S STATUS REGISTER
 4510                                                                  ;FAILED TO SET WHEN BIT13 WAS SET
 4511                                                                  ;AND A MAINTENANCE ST1 GENERATED.
 4512
 4513
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 4517  011640  005077  167460          1$:     CLR     @ASR            ;LEAVE SUBTEST WITH CLOCK CLEAR.
 4518
 4519
 4520                                   ;;*****************************************************************
 4521                                   ;*TEST 111      *TEST THAT CLOCK A WILL INCREMENT - MODE 0 - RATE STP1 FIRST COUNT TEST
 4522                                   ;*
 4523                                   ;*COUNT TEST - THIS IS THE VERY FIST TIME THAT THE COUNTER
 4524                                   ;*HAS BEEN ASKED TO INCREMENT! WHAT WE ARE GOING TO DO IS
 4525                                   ;*CLEAR THE BUFFER, SELECT MODE 0, RATE OF STP1.
 4526                                   ;*NEXT WILL GENERATE THE STP1 THROUGH MAINTENANCE MODE (WE'VE
 4527                                   ;*DONE THIS BEFORE IN A PREVIOUS TEST TO SEE IF THE FLAG WOULD SET)
 4528                                   ;*AND SEE IF THE COUNTER HAS INCREMENTED.
 4529                                   ;*
 4530                                   ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
```

```
4531                                    ;*
4532                                    ;*
4533                                    ;;*****************************************************************
4534  011644  000004                    TST111: SCOPE
4535
4536  011646  005077  167452                   CLR     @ASR            ;CLEAR CLOCK A.
4537  011652  005077  167450                   CLR     @ABR            ;CLEAR BUFFER REGISTER.
4538  011656  012777  000015  167440          MOV     #15,@ASR        ;SELECT: MODE0! RATE "STP1" AND ENABLE
4539                                                                    ;CLOCK A TO COUNT.
4540  011664  052777  010000  167432          BIS     #BIT12,@ASR     ;NOW GENERATE A MAINTENANCE STP1 - AT
4541                                                                    ;THIS TIME THE CLOCK SHOULD COUNT ONCE.
4542  011672  012737  000001  001124          MOV     #1,$GDDAT       ;FOR ERROR TYPEOUT (IF NEEDED) SET THE #1.
4543  011700  017737  167424  001126          MOV     @ACR,$BDDAT     ;READ THE COUNT REGISTER - SHOULD=1.
4544  011706  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THE COUNTER COUNT ONCE?
4545  011714  001401                          BEQ     1$              ;IF YES - BR NEXT TEST.
4546
4547

          ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

4551  011716  104011                    ERROR   11              ;ERROR - COUNT A FAILED TO COUNT
4552                                                            ;ONCE, WHEN ENABLED, MODE 0
4553                                                            ;RATE "STP1" SEE ABOVE COMMENTS
4554                                                            ;FOR COMPLETE DESCRIPTION AND LIST
4555                                                            ;OF EVENTS.
4556
4557

          ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

4561  011720                    1$:
4562
4563
4564                                    ;;*****************************************************************
4565                                    ;*TEST 112      *TEST THE ABILITY OF CLOCK A TO COUNT FROM ZERO TO OVERFLOW USING M STP1
4566                                    ;*
4567                                    ;*IN THIS TEST WE'LL COUNT THE COUNTER THROUGH EACH STEP
4568                                    ;*FROM ZERO TO OVERFLOW USING MAINTENANCE "STP1" COUNTS.
4569                                    ;*IT IS KNOWN THAT THE COUNTER WILL INCREMENT ONCE USING
4570                                    ;*THE MAINTENANCE STP1 AND THAT THE COUNTER F/FS WILL
4571                                    ;*PASS ALL DATA BITS.
4572                                    ;*UNKNOW IS THE ABILITY OF THE F/F'S TO PROPAGATE THEIR
4573                                    ;*OVERFLOWS.
4574                                    ;*
4575                                    ;*IF IT IS DESIRED TO START THIS TEST AT A VALUE OTHER THAN
4576                                    ;*ZERO, CHANGE THE SECOND INSTR. OF THIS TEST TO A VALUE TO BE
4577                                    ;*LOADED INTO THE BUFFER TO THAT VALUE DESIRED.
4578                                    ;*
4579                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
4580                                    ;*
4581                                    ;*
4582                                    ;;*****************************************************************
4583  011720  000004                    TST112: SCOPE
4584  011722  012737  000001  001164          MOV     #1,$TIMES       ;;DO 1 ITERATION
4585
4586  011730  012737  000000  001124          MOV     #0,$GDDAT       ;START THE COUNTER FROM ZERO.
```

```
4587
4588
4589
4590  011736  005077  167362          1$:   CLR   @ASR            ;NOTE: A VALUE OTHER THAN ZERO MAY BE
4591  011742  013777  001124  167356        MOV   $GDDAT,@ABR      ;PATCHED IN HERE IN ORDER THAT A COUNT
4592                                                               ;MAY BE STARTED HIGHER.
4593                                                               ;DISABLE CLOCK A.
4594                                                               ;LOAD THE COUNTER BUFFER WITH VALUE.
4595                                                               ;"1$" IS THE LOOP BACK POINT ON
4596  011750  012777  000015  167346        MOV   #15,@ASR         ;"LOOP ON TEST" (SW14=1) FEATURE. NORMAL
4597                                                               ;LOOP BACK POINT WILL BE "2$".
4598  011756  052777  010000  167340  2$:   BIS   #BIT12,@ASR      ;ENABLE COUNTER TO COUNT. SELECTED:
4599                                                               ;MODE 0, RATE "STP1"
4600                                                               ;GENERATE A MAINTENANCE "STP1". THIS
4601  011764  005237  001124              INC   $GDDAT           ;SHOULD CAUSE THE CLOCK TO INCREMENT
4602                                                               ;ONE VALUE.
4603                                                               ;$GDDAT IS USED TO KEEP TRACK OF THE
4604  011770  017737  167334  001126        MOV   @ACR,$BDDAT      ;VALUE THE CLOCK SHOULD COUNT TO.
4605
4606  011776  023737  001126  001124        CMP   $BDDAT,$GDDAT    ;READ THE CLOCKS COUNT REGISTER.
4607  012004  001402                        BEQ   3$               ;DID COUNT OCCUR CORRECTLY?
4608                                                               ;IF YES - BR TO "3$".
4609

          ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

4613  012006  104011                    ERROR  11              ;ERROR - CLOCK A FAILED TO UPCOUNT
4614                                                           ;CORRECTLY USING MODE 0 - RATE "STP1".
4615
4616

          ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

4620  012010  000403                    BR     4$
4621  012012  005737  001124      3$:   TST    $GDDAT          ;DID WE ACHIEVE OVERFLOW TO ZERO YET?
4622  012016  001410                    BEQ    5$              ;IF YES - BR NEXT TEST
4623
4624  012020  032777  040000  167110  4$:   BIT    #BIT14,@SWR     ;LOOP CURRENT COUNT?
4625  012026  001753                    BEQ    2$              ;BR IF NO TO NEXT COUNT UPDATE.
4626  012030  162737  000001  001124        SUB    #1,$GDDAT       ;IF YES - DECREMENT EXPECTED AND RELOAD.
4627  012036  000737                    BR     1$              ;GOTO RELOAD POINT.
4628
4629  012040                      5$:                           ;END SUBTEST
4630
4631                              .SBTTL  *
4632                              .SBTTL  * PHASE 3 CLOCK A COUNT FUNCTION TESTS
4633                              .SBTTL  *
4634
4635
4636                              ;;**************************************************************
4637                              ;*TEST 113      *TEST THAT CLOCK A OVERFLOW WILL OCCUR
4638                              ;*
4639                              ;*NOW WE'LL TRY AND GENERATE "A OVERFLOW L"
4640                              ;*WHICH SETS BDO5. WE'LL DO IT BY PRESETTING THE BUFFER
4641                              ;*AT 177777 AND GENERATE A MAINTENANCE STP1. WE ALREADY
4642                              ;*KNOW MAINTENANCE STP1'S WILL ADVANCE THE COUNTER.
```

```
                                        ;*ALSO SEE IF "MODE" FLG GETS SET.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************************
012040  000004                          TST113: SCOPE

012042  005077  167256                  CLR     @ASR            ;MAKE SURE CLOCK A CLEAR.
012046  012777  177777  167252          MOV     #177777,@ABR    ;PRESET BUFFER TO ALL ONES.
012054  012777  000015  167242          MOV     #15,@ASR        ;SELECT MODE 0; RATE STP4; GO.
012062  052777  010000  167234          BIS     #BIT12,@ASR     ;GENERATE A MAINTENANCE STP1.

012070  032777  000040  167226  1$:     BIT     #BIT05,@ASR     ;DID OVERFLOW BIT SET?
012076  001002                          BNE     2$              ;BR IF YES TO "2$".

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

012100  104012                          ERROR   12              ;ERROR BIT05 OF CSR CLOCK A FAILED
                                                                ;TO SET ON OVERFLOW.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

012102  000405                          BR      3$

012104  032777  000200  167212  2$:     BIT     #BIT07,@ASR     ;DID BIT07 - "MODE" FLG SET?
                                                                ;IT SHOULD SET WHEN BIT05 SETS.
012112  001001                          BNE     3$              ;BR IF YES TO 3$.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

012114  104012                          ERROR   12              ;OVERFLOW FAILED TO SET CLOCK A'S
                                                                ;CSR BIT07 "MODE" FLG. IT
                                                                ;HAD; HOWEVER SET BIT05 "OVERFLOW"
                                                                ;FLAG.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

012116                          3$:
                                ;;*****************************************************************
                                ;*TEST 114      *TEST IN CLOCK A THAT OVERFLOW IN MODE 0 CAUSE CLEARING OF "ENB CNTR" F/
                                ;*
                                ;*WE'RE GOING TO SEE IF "A RELOAD" H AND "MODE 0" H CLEAR
                                ;*"ENB CNTR A" F/F.  "A RELOAD" GENERATED ON OVERFLOW AND
                                ;*"MODE 0" GENERATED BY "MODE A0 (0)" AND "MODE A0 (1)"
                                ;*ALSO SEE IF COUNTER REGISTER GETS RELOADED
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                ;*
                                ;*
                                ;;*****************************************************************
012116  000004                  TST114: SCOPE
```

```
4699  012120  005077  167200              CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
4700  012124  012777  177777  167174      MOV     #177777,@ABR    ;PRESET BUFFER TO ALL ONES.
4701  012132  012777  000015  167164      MOV     #15,@ASR        ;SELECT MODE 0, RATE "STP4", GO.
4702  012140  052777  010000  167156      BIS     #BIT12,@ASR     ;GENERATE A MAINTENANCE STPI.
4703                                                              ;THIS SHOULD CAUSE AN OVERFLOW
4704                                                              ;OVERFLOW WILL GENERATE "A RELOAD" H
4705                                                              ;COMBINE THIS WITH MODE 0 AND WE
4706                                                              ;SHOULD CLEAR "ENB CNTR A" F/F.
4707
4708  012146  032777  000001  167150      BIT     #BIT00,@ASR     ;DID BIT00 "ENB CNTR A" F/F CLEAR?
4709  012154  001402                      BEQ     1$              ;BR IF YES - NEXT TEST.
4710
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4714  012156  104012                      ERROR   12              ;"ENB CNTR A" F/F NOT CLEARED
4715                                                              ;ON OVERFLOW.
4716
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4720  012160  000405                      BR      2$
4721
4722  012162  022777  177777  167140  1$: CMP     #177777,@ACR    ;DID COUNTER GET RELOADED AFTER
4723                                                              ;THE OVERFLOW OCCURRED?
4724  012170  001401                      BEQ     2$              ;BR IF YES - NEXT TEST.
4725
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4729  012172  104012                      ERROR   12              ;ERROR CLOCK A - COUNTER DID NOT GET RELOAD FROM
4730                                                              ;BUFFER AFTER OVERFLOW. "A RELOAD" H
4731                                                              ;DID GET GENERATED ON WE WOULD
4732                                                              ;HAVE GOT PREVIOUS ERROR; THEREFORE PROBLEM
4733                                                              ;MUST EXIST WITH COMBINING "A RELOAD" H
4734                                                              ;WITH "MODE A1 (0)" H TO GENERATE A
4735                                                              ;COUNTER LOAD.
4736
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

4740  012174                        2$:
4741
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 115      *TEST THE ABILITY OF CLOCK A TO COUNT AT 1MHZ RATE PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
                                        ;*IN RATE: 1MHZ          PART 1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
012174  000004                          TST115: SCOPE
012176  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

012204  005077  167114                          CLR     @ASR            ;/MAKE SURE CLOCK IS CLEAR.
012210  005077  167112                          CLR     @ABR            ;/CLEAR THE BUFFER.
012214  012777  000003  167102                  MOV     #1!2,@ASR       ;/SELECT: MODE 0, RATE 1MHZ ; GO.
012222  005000                                  CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
012224  005200                          1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
012226  001376                                  BNE     1$
012230  005777  167074                          TST     @ACR            ;/DID COUNTER INCREMENT AT ALL?
012234  001005                                  BNE     2$              ;/IF YES - BR NEXT TEST.

012236  032777  000040  167060                  BIT     #BIT05,@ASR     ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
012244  001001                                  BNE     2$              ;/BR IF YES NEXT TEST.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012246  104012                                  ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: 1MHZ.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012250                                  2$:
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 116      *TEST THE ABILITY OF CLOCK A TO COUNT AT 100KHZ RATE PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
                                        ;*IN RATE: 100KHZ                PART 1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
012250  000004                          TST116: SCOPE
012252  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

012260  005077  167040                          CLR     @ASR            ;/MAKE SURE CLOCK IS CLEAR.
012264  005077  167036                          CLR     @ABR            ;/CLEAR THE BUFFER.
012270  012777  000005  167026                  MOV     #1!4,@ASR       ;/SELECT: MODE 0, RATE 100KHZ ; GO.
012276  005000                                  CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
012300  005200                          1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
012302  001376                                  BNE     1$
012304  005777  167020                          TST     @ACR            ;/DID COUNTER INCREMENT AT ALL?
012310  001005                                  BNE     2$              ;/IF YES - BR NEXT TEST.

012312  032777  000040  167004                  BIT     #BIT05,@ASR     ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
012320  001001                                  BNE     2$              ;/BR IF YES NEXT TEST.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012322  104012                                  ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: 100KHZ.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012324                                  2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 117      *TEST THE ABILITY OF CLOCK A TO COUNT AT 10KHZ RATE PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
                                        ;*IN RATE: 10KHZ                    PART 1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
012324  000004                    TST117: SCOPE
012326  012737  000020  001164            MOV     #20,$TIMES      ;;DO 20 ITERATIONS

012334  005077  166764                    CLR     @ASR            ;/MAKE SURE CLOCK IS CLEAR.
012340  005077  166762                    CLR     @ABR            ;/CLEAR THE BUFFER.
012344  012777  000007  166752            MOV     #1!6,@ASR       ;/SELECT: MODE 0, RATE 10KHZ ; GO.
012352  005000                            CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
012354  005200                    1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
012356  001376                            BNE     1$
012360  005777  166744                    TST     @ACR            ;/DID COUNTER INCREMENT AT ALL?
012364  001005                            BNE     2$              ;/IF YES - BR NEXT TEST.

012366  032777  000040  166730            BIT     #BIT05,@ASR     ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
                                                                  ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                  ;/F/F HAD SET.
012374  001001                            BNE     2$              ;/BR IF YES NEXT TEST.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

012376  104012                            ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                  ;/COUNT RATE: 10KHZ.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

012400                          2$:
```

```
4859                                                    ;/*
4860
4861                                            ;;**********************************************************************
4862                                            ;*TEST 120      *TEST THE ABILITY OF CLOCK A TO COUNT AT 1KHZ RATE PART 1
4863                                            ;*
4864                                            ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
4865                                            ;*IN RATE: 1KHZ          PART 1
4866                                            ;*
4867                                            ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
4868                                            ;*
4869                                            ;*
4870                                            ;;**********************************************************************
4871  012400  000004                            TST120: SCOPE
4872  012402  012737  000020  001164                    MOV     #20,$TIMES      ;;DO 20 ITERATIONS
4873
4874  012410  005077  166710                            CLR     @ASR            ;/MAKE SURE CLOCK IS CLEAR.
4875  012414  005077  166706                            CLR     @ABR            ;/CLEAR THE BUFFER.
4876  012420  012777  000011  166676                    MOV     #1!10,@ASR      ;/SELECT: MODE 0, RATE 1KHZ ; GO.
4877  012426  005000                                    CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
4878  012430  005200                            1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
4879  012432  001376                                    BNE     1$
4880  012434  005777  166670                            TST     @ACR            ;/DID COUNTER INCREMENT AT ALL?
4881  012440  001005                                    BNE     2$              ;/IF YES - BR NEXT TEST.
4882
4883  012442  032777  000040  166654                    BIT     #BIT05,@ASR     ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
4884                                                                            ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
4885                                                                            ;/F/F HAD SET.
4886  012450  001001                                    BNE     2$              ;/BR IF YES NEXT TEST.
4887
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4891  012452  104012                                    ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
4892                                                                            ;/COUNT RATE: 1KHZ.
4893
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4897  012454                                    2$:
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 121      *TEST THE ABILITY OF CLOCK A TO COUNT AT 100HZ RATE PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
                                        ;*IN RATE: 100HZ                  PART 1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
012454  000004                          TST121: SCOPE
012456  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

012464  005077  166634                          CLR     @ASR            ;/MAKE SURE CLOCK IS CLEAR.
012470  005077  166632                          CLR     @ABR            ;/CLEAR THE BUFFER.
012474  012777  000013  166622                  MOV     #1!12,@ASR      ;/SELECT: MODE 0, RATE 100HZ ; GO.
012502  005000                                  CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
012504  005200                          1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
012506  001376                                  BNE     1$
012510  005777  166614                          TST     @ACR            ;/DID COUNTER INCREMENT AT ALL?
012514  001005                                  BNE     2$              ;/IF YES - BR NEXT TEST.

012516  032777  000040  166600                  BIT     #BIT05,@ASR     ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
012524  001001                                  BNE     2$              ;/BR IF YES NEXT TEST.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012526  104012                                  ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: 100HZ.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012530                                  2$:
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 122      *TEST THE ABILITY OF CLOCK A TO COUNT AT LINE-FREQ RATE PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
                                        ;*IN RATE: LINE-FREQ            PART 1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
012530  000004                          TST122: SCOPE
012532  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

012540  005077  166560                          CLR     @ASR            ;/MAKE SURE CLOCK IS CLEAR.
012544  005077  166556                          CLR     @ABR            ;/CLEAR THE BUFFER.
012550  012777  000017  166546                  MOV     #1!16,@ASR      ;/SELECT: MODE 0, RATE LINE-FREQ ; GO.
012556  005000                                  CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
012560  005200                          1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
012562  001376                                  BNE     1$
012564  005777  166540                          TST     @ACR            ;/DID COUNTER INCREMENT AT ALL?
012570  001005                                  BNE     2$              ;/IF YES - BR NEXT TEST.

012572  032777  000040  166524                  BIT     #BIT05,@ASR     ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
012600  001001                                  BNE     2$              ;/BR IF YES NEXT TEST.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012602  104012                                  ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: LINE-FREQ.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

012604                                  2$:


                                        ;;*****************************************************************
                                        ;*TEST 123      *TEST THAT CLOCK A DOESN'T COUNT WHEN NO RATE IS SELECTED
                                        ;*
                                        ;*WE KNOW THAT CLOCK A COUNTS AT ALL
                                        ;*ITS RATES, SO LETS BE SURE IT DOESNT
                                        ;*COUNT WHEN NO RATE IS SELECTED. ON
                                        ;*ERROR SUSPECT DUAL RATE SELECTION.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************************
012604  000004                          TST123: SCOPE
012606  012737  000050  001164                  MOV     #50,$TIMES      ;;DO 50 ITERATIONS

012614  005077  166504                          CLR     @ASR            ;CLEAR CLOCK A.
```

```
 4993  012620  005077  166502                CLR     @ABR            ;ZERO ITS BUFFER.
 4994  012624  005277  166474                INC     @ASR            ;TELL THE CLOCK TO GO!
 4995                                                                 ;NO RATE SELECTED.
 4996  012630  005000                        CLR     R0
 4997  012632  005200                1$:     INC     R0
 4998  012634  001376                        BNE     1$
 4999  012636  005777  166466              TST     @ACR            ;ANY COUNT OCCUR?
 5000  012642  001004                        BNE     2$              ;IF COUNTER HAS SOMETHING IN IT REPORT ERROR
 5001
 5002  012644  032777  000040  166452      BIT     #BIT05,@ASR     ;IF THE OVERFLOW BIT SET THEN IT MUST
 5003                                                                 ;HAVE COUNTED.
 5004  012652  001401                        BEQ     3$
 5005
 5006
                ;;;$$$
```

```
 5049  012712  001401                          BEQ     1$              ;BR IF NO - NEXT TEST.
 5050
                ;;;$$$
```

```
                                        ;*IS ["MODE A1 (0)" H + "A RELOAD" H]. WE'LL STILL BE GENERATING
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*"A RELOAD" H BUT SHOULD NOT GET."MODE A1 (0)" H AS THIS IS MODE 2.
                                        ;*
                                        ;;*****************************************************************
012760  000004                          TST126: SCOPE

012762  005077  166336                          CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
012766  012777  177777  166332                  MOV     #177777,@ABR    ;SET BUFFER + COUNT REG TO -1 FROM OVERFLOW.
012774  012777  001015  166322                  MOV     #1015,@ASR      ;SET: MODE 2; RATE STP1; GO.
013002  052777  010000  166314                  BIS     #BIT12,@ASR     ;CAUSE A MAINTENANCE STP1 - CLOCK ONCE.
                                                                        ;THIS SHOULD CAUSE AN OVERFLOW AND LEAVE
                                                                        ;THE COUNT REGISTER CLEAR AS A
                                                                        ;BUFFER TO COUNT REG. SHOULDN'T OCCUR.
013010  005777  166314                          TST     @ACR            ;IS THE COUNT REG. CLEAR?
013014  001401                                  BEQ     1$              ;BR IF YES TO NEXT TEST.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

013016  104012                                  ERROR   12              ;ERROR THE CONTENTS OF THE BUFFER REG.
                                                                        ;GOT TRANSFERRED TO THE COUNT REG.
                                                                        ;(CLOCK A) ON OVERFLOW DOING A
                                                                        ;MODE 2 OPERATION.

                                                                        ;THERE'S AN OUTSIDE CHANCE THAT THE
                                                                        ;COUNT REG. NEVER GOES TO ZERO ON
                                                                        ;AN OVERFLOW - THIS IS THE FIRST TIME
                                                                        ;THAT WE WERE ABLE TO LOOK AT IT ON
                                                                        ;OVERFLOW BECAUSE MODES 0 + 1 CAUSED
                                                                        ;THAT AUTOMATIC BUFFER TO COUNT REG.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

013020                                  1$:


                                        ;;*****************************************************************
                                        ;*TEST 127       *TEST THAT CLOCK A MODE 2 + MAINTENANCE ST2 SET MODE FLG
                                        ;*
                                        ;*NOW WE'LL SEE IF CAN GENERATE A "CNTR TO BUFF" H SIGNAL.
                                        ;*TO DETECT IT, WE'RE GOING TO DEPEND ON IT SETTING THE MODE FLAG,
                                        ;*CLOCK A CSR BIT07. ["MODE A1 (0)" H + "ST2 (1)" H] + "TP0" L="CNTR TO BUFF" H.
                                        ;*BEING IN MODE 2, SHOULD GIVE US "MODE A1 (1)" H. WELL GET ST2 (1) H
                                        ;*BY GENERATING A MAINTENANCE "ST2". TP0 COMES FROM THE INTERNAL
                                        ;*CLOCK PAGE.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "RD STAT B"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
013020  000004                          TST127: SCOPE
```

```
5161  013022  005777  166304          TST     @BSR            ;GENERATE A SYNC PULSE
5162  013026  005077  166272          CLR     @ASR            ;MAKE SURE CLOCK A'S STAT REG IS CLEAR.
5163  013032  012777  001000  166264  MOV     #1000,@ASR      ;SET MODE 2.
5164  013040  052777  002000  166256  BIS     #BIT10,@ASR     ;GENERATE MAINTENANCE ST2.
5165                                                          ;THE COMBO OF MODE 2 + ST2 SHOULD
5166                                                          ;GET "CNTR TO BUFF" H WHICH SHOULD
5167                                                          ;SET "MODE FLG" F/F.
5168  013046  032777  000200  166250  BIT     #BIT07,@ASR     ;DID IT SET?
5169  013054  001001                  BNE     1$              ;IF YES-BR TO NEXT TEST.
5170
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5174  013056  104012                  ERROR   12              ;ERROR - MODE 2 + ST2 DID NOT SET MODE FL.
5175
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5179  013060                  1$:

5180
5181
5182                          ;;*****************************************************************
5183                          ;*TEST 130      *TEST THAT PATTERN 052525 CAN BE XFERRED BETWEEN A'S COUNT-BUFFER REGS
5184                          ;*
5185                          ;*NOW WE'LL SHOT THE WORKS - WE KNOW FROM THE PREVIOUS TEST WE
5186                          ;*CAN GENERATE "CNTR TO BUFF" H FROM MODE 2 + MAINTENANCE ST2, NOW
5187                          ;*WE WELL TRY AND GENERATE A TRANSFER BETWEEN THE COUNTER AND BUFFER
5188                          ;*USING A CB PAT PATTERN.
5189                          ;*IF NO DATA PATTERN GETS TRANSFERRED, SUSPECT SIG "LD BUFFER"
5190                          ;*TO BE STUCK LOW AT THE MUX INPUT FOR THE BUFFER OR
5191                          ;*"CNTR TO BUFF" H NOT GETTING THROUGH TO THE LOAD INPUTS OF THE
5192                          ;*BUFFER REGISTER.
5193                          ;*IF JUST ONE OR A FEW BITS GETS MESSED UP ON THE XFERR-
5194                          ;*SUSPECT THE RESPECTIVE MUX OR ETCH BETWEEN THE COUNT REG
5195                          ;*AND MUX. GOOD LUCK.
5196                          ;*
5197                          ;* PROBABLE SYNC POINT FOR THIS TEST:: "RD STAT B"
5198                          ;*
5199                          ;*
5200                          ;*****************************************************************
5201  013060  000004          TST130: SCOPE
5202
5203  013062  005777  166244          TST     @BSR            ;GENERATE A SYNC PULSE
5204  013066  005077  166232          CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
5205  013072  012777  052525  166226  MOV     #052525,@ABR    ;PUT PATTERN 052525 INTO BUFFER REG.
5206                                                          ;IT SHOULD GET XFERRED TO COUNT REG.
5207  013100  012777  001001  166216  MOV     #1001,@ASR      ;SELECT: MODE 2, ENABLE.
5208  013106  005077  166214          CLR     @ABR
5209  013112  052777  002000  166204  BIS     #BIT10,@ASR     ;NOW GENERATE A MAINTENANCE ST2.
5210  013120  012737  052525  001124  MOV     #052525,$GDDAT  ;RECORD $GDDAT (PATTERN) IN CASE WE
5211                                                          ;NEED TO TYPE OUT AN ERROR.
5212  013126  017737  166174  001126  MOV     @ABR,$BDDAT     ;NOW READ BACK THE BUFFER REG.
5213  013134  023737  001126  001124  CMP     $BDDAT,$GDDAT   ;WAS THE TRANSFER SUCCESSFUL?
5214  013142  001401                  BEQ     1$              ;IF YES THEN BR TO NEXT TEST.
5215
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
5219  013144  104012                          ERROR   12              ;ERROR FAILED TO XFERR 052525 PATTERN
5220                                                                  ;CORRECTLY FROM COUNT TO BUFFER REG.
5221                                                                  ;SEE INIT. COMMENT AS TO WHY
5222                                                                  ;IT MIGHT HAVE GONE SOUR.
5223
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5227  013146                        15:
5228          000011                        P=P+1
5229
5230                                ;;;
```

```
013234                                  1$:
        000012                                  P=P+1


                                        ;;*****************************************************************
                                        ;*TEST 132      *TEST THAT A'S COUNT REG. IS CLEARED BY INIT
                                        ;*
                                        ;*HERE'S A QUICKY LITTLE TEST DESIGNED TO BE SURE THAT A'S
                                        ;*COUNT REGISTER CAN BE CLEARED BY INIT.
                                        ;*
                                        ;;*****************************************************************
013234  000004                          TST132: SCOPE
013236  012737  000005  001164                  MOV     #5,$TIMES       ;;DO 5 ITERATIONS

013244  005077  166054                          CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
013250  012777  177777  166050                  MOV     #177777,@ABR    ;LOAD ALL ONE'S INTO BUFF +
                                                                        ;COUNT REG.
013256  005037  001124                          CLR     $GDDAT          ;FIX $GDDAT FOR ERROR TYPE OUT - IF ANY.

013262  000005                                  RESET                   ;SYSTEM INITIALIZE - SHOULD CLEAR
                                                                        ;COUNT REG.
013264  017737  166040  001126                  MOV     @ACR,$BDDAT     ;READ COUNT REG. - SHOULD BE CLEAR.
013272  001401                                  BEQ     1$              ;IF YES - BR NEXT TEST.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

013274  104004                                  ERROR   4               ;ERROR - SYS. INIT. FAILED TO CLEAR
                                                                        ;COUNT REG.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

013276                                  1$:


                                        ;;*****************************************************************
                                        ;*TEST 133      *TEST THAT A'S COUNT REGISTER ISN'T CLEARED IN MODE 1 WHEN STP2 IS GENER
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO MAKE SURE THAT WE DON'T CLEAR THE COUNT
                                        ;*REGISTER IN MODE 1 WHEN AN STP2 IS GENERATED.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
013276  000004                          TST133: SCOPE

013300  005077  166020                          CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
013304  012777  177777  166014                  MOV     #177777,@ABR    ;LOAD ALL ONES INTO BUFFER COUNT REGS.
013312  012777  000200  166004                  MOV     #200,@ASR       ;SET MODE 1.
013320  052777  002000  165776                  BIS     #BIT10,@ASR     ;GENERATE A MAINTENANCE STP2.
013326  005777  165776                          TST     @ACR            ;SEE IF IT CLEARED COUNT REG.
013332  001001                                  BNE     1$              ;BR IF NO - NEXT TEST.
```

```
5329
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

5333  013334  104012                          ERROR   12              ;ERROR CLOCK A MODE 1 + STP2 CLEARED COUNT REG.
5334                                                                  ;TO SCOPE FIND OUT WHAT IS GENERATING
5335                                                                  ;SIGNAL ON CLR INPUT OF COUNT REG.
5336
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

5340  013336                          1$:
5341
5342                                    ;;**********************************************************************
5343                                    ;*TEST 134      *TEST THAT A'S COUNT REGISTER ISN'T CLEARED IN MODE 2 WHEN STP2 IS GENER
5344                                    ;*
5345                                    ;*THIS TEST IS DESIGNED TO MAKE SURE THAT WE DON'T CLEAR THE COUNT
5346                                    ;*REGISTER IN MODE 2 WHEN AN STP2 IS GENERATED.
5347                                    ;*
5348                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
5349                                    ;*
5350                                    ;*
5351                                    ;;**********************************************************************
5352  013336  000004                    TST134: SCOPE
5353
5354  013340  005077  165760                    CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
5355  013344  012777  177777  165754            MOV     #177777,@ABR    ;LOAD ALL ONES INTO BUFFER COUNT REGS.
5356  013352  012777  001000  165744            MOV     #1000,@ASR      ;SET MODE 2.
5357  013360  052777  002000  165736            BIS     #BIT10,@ASR     ;GENERATE A MAINTENANCE STP2.
5358  013366  005777  165736                    TST     @ACR            ;SEE IF IT CLEARED COUNT REG.
5359  013372  001001                            BNE     1$              ;BR IF NO - NEXT TEST.
5360
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

5364  013374  104012                          ERROR   12              ;ERROR CLOCK A MODE 2 + STP2 CLEARED COUNT REG.
5365                                                                  ;TO SCOPE FIND OUT WHAT IS GENERATING
5366                                                                  ;SIGNAL ON CLR INPUT OF COUNT REG.
5367
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSS

5371  013376                          1$:
5372
5373
5374                                    ;;**********************************************************************
5375                                    ;*TEST 135      *TEST THAT MODE 3 + "STP2" CLEARS A'S COUNT REGISTER.
5376                                    ;*
5377                                    ;*IN THIS TEST WE'LL DO A MODE 3 FOR THE FIRST TIME.
5378                                    ;*MODE 3 + "STP2" SHOULD CLEAR THE COUNT REGISTER.
5379                                    ;*WE KNOW FROM A PREVIOUS TEST THAT "INIT" L WAS ABLE TO
5380                                    ;*CLEAR THE REG. AND ALSO THAT WE CAN GENERATE AN "STP2" H.
5381                                    ;*"MODE A0 (1)" H + "MODE A1 (1)" H + "STP2" H = CLEARING SIGNAL.
5382                                    ;*
5383                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
5384                                    ;*
```

```
                                        ;*
                                        ;;**********************************************************************
013376  000004                        TST135: SCOPE

013400  012777  001400  165716          MOV     #1400,@ASR      ;CLEAR CLOCK A + SELECT MODE 3.
013406  012777  177777  165712          MOV     #177777,@ABR    ;LOAD ALL ONE'S TO BUFFER + COUNTER REGS.
013414  052777  002000  165702          BIS     #BIT10,@ASR     ;GENERATE A MAINTENANCE "STP2".
                                                                ;THIS SHOULD CLEAR THE COUNT REG.
013422  005777  165702                  TST     @ACR            ;IS COUNT REG. CLEAR?
013426  001401                          BEQ     1$              ;BR IF YES - NEXT TEST.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

013430  104012                          ERROR   12              ;ERROR-CLOCK A-COUNT REGISTER FAILED TO
                                                                ;CLEAR IN MODE 3 ON "STP2" PULSE.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

013432                          1$:


                                        ;;**********************************************************************
                                        ;*TEST 136      *TEST THE AUTODECREMENT FEATURE OF CLOCK A'S BUFFER
                                        ;*
                                        ;*THIS IS GOING TO BE A BIGGIE-WITH TWO PARTS.
                                        ;*PART1:        WE'RE LOADING THE BUFFER WITH ALL ONES; MODE 0; RATE STP1,
                                        ;*              WITH "AUTO INC (1)" SET (FOR FIRST TIME!). NOW WE'LL GENERATE
                                        ;*              THE THING TO WATCH FOR IS THE EXPLOSIVE COMBO OF
                                        ;*              "MODE A0 (0)" H + "MODE A1 (0)" H (MODE 0) + "AUTO INC (1)" H +
                                        ;*              "A OVERFLOW" ALL FEEDING THE DOWN COUNT SIDE OF THE 74193 CHIP.
                                        ;*              THE RESULTS SHOULD BE 177776.
                                        ;*PART2:        IF PART 1 WAS SUCCESSFUL WE'LL LOOK TO SEE IF
                                        ;*              THE COUNTER GOT LOADED WITH THE NEW VALUE 177776.
                                        ;*              THE HANG-UP HERE IS THAT "A OVERFLOW" FEEDS A ONE SHOT
                                        ;*              THAT GENERATES "A RELOAD" H. WE KNOW THAT WE CAN
                                        ;*              GET "A RELOAD" H BUT THE BIG TEST HERE IS SEEING IF THAT
                                        ;*              ONE SHOT SPITS OUT "A RELOAD" H TOO SOON TO CATCH THE BUFFER
                                        ;*              WITH ITS PANTS DOWN AS ITS DECREMENTING ITSELF.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;**********************************************************************
013432  000004                        TST136: SCOPE

013434  005077  165672                  CLR     @BSR            ;MAKE SURE CLOCK B IS CLEAR.
013440  005077  165660                  CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
013444  012777  177777  165654          MOV     #177777,@ABR    ;LOAD BUFFER + COUNT REGS.
013452  012777  000020  165652          MOV     #20,@BSR        ;SET AUTO INCREMENT MODE
013460  012777  000015  165636          MOV     #15,    @ASR    ;ENABLE COUNTER, MODE 0, RATE STP1
013466  052777  010000  165630          BIS     #BIT12,@ASR     ;ZAP! A MAINTENANCE STP1 HAS BEEN MADE.
                                                                ;STOP HERE AND LOOK WHAT JUST HAPPENED.
                                                                ;1. STP1 CLOCKED THE COUNTER (SET 177777).
```

```
5441                                                    ;2. COUNTER OVERFLOWED - "A OVERFLOW" L
5442                                                    ;3. MODE 0 + "A OVERFLOW" L + "A AUTO INC (1)" H
5443                                                    ;   DOWN COUNTED THE BUFFER.
5444                                                    ;4. "A OVERFLOW" L GOES THROUGH ONE SHOT DELAY
5445                                                    ;   TO GIVE "A RELOAD" H
5446                                                    ;5. "A RELOAD" H CAUSES A BUFFER TO COUNTER
5447                                                    ;   RELOAD.
5448
5449                                    ;PART 1 DID BUFFER GET DECREMENTED?
5450
5451  013474  012737  177776  001124          MOV     #177776,$GDDAT  ;SET FOR ERROR TYPEOUT IF ANY.
5452  013502  017737  165620  001126          MOV     @ABR,$BDDAT     ;READ THE RESULTS OF THE BUFFER.
5453  013510  023727  001126  177776          CMP     $BDDAT,#177776  ;DID BUFFER DECREMENT TO 177776?
5454  013516  001402                          BEQ     1$              ;BR IF YES TO PART 2
5455
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5459  013520  104012                          ERROR   12              ;ERROR-CLOCK A. AFTER AN OVERFLOW
5460                                                                  ;WITH AUTO INC ENABLED MODE 0, THE
5461                                                                  ;BUFFER REGISTER FAILED TO DOWN COUNT
5462                                                                  ;SEE ABOVE COMMENTS FOR SEQUENCE
5463                                                                  ;OF EVENTS.
5464  013522  000410                          BR      2$              ;IF ERROR ONLY LOOP ON PART 1.
5465
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5469  013524                          1$:     ;PART 2 DID NEW COUNT GET TRANSFERRED TO COUNT REGISTER?
5470
5471  013524  017737  165600  001126          MOV     @ACR,$BDDAT     ;READ THE COUNT REGISTER
5472  013532  023727  001126  177776          CMP     $BDDAT,#177776  ;DID NEW VALUE OF THE BUFFER
5473                                                                  ;REGISTER GET PROPERLY LOADED INTO
5474                                                                  ;THE COUNT REGISTER?
5475  013540  001401                          BEQ     2$              ;BR IF YES - NEXT TEST.
5476
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5480  013542  104012                          ERROR   12              ;ERROR CLOCK A. NEW CONTENTS OF
5481                                                                  ;BUFFER REGISTER FAILED TO BE PROPERLY
5482                                                                  ;LOADED INTO COUNT REGISTER AFTER
5483                                                                  ;AUTO DECREMENT.
5484                                                                  ;SEE ABOVE COMMENTS FOR SEQUENCE OF EVENTS.
5485
            ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5489  013544  005077  165562          2$:     CLR     @BSR            ;CLEAR AUTO INC OPTION.
5490
5491                                    ;;*****************************************************************
5492                                    ;*TEST 137      *TEST THAT CLOCK A'S 1 MHZ CLK CAN BE DISABLED
5493                                    ;*
5494                                    ;*FOR THIS TEST, WE'LL TRY DISABLING THE 1 MHZ CLOCK. TO DO THIS,
5495                                    ;*WE'LL SET THE "DISABLE OSC 1 MHZ" BIT 11 IN CLOCK B SR. THEN
5496                                    ;*COUNTED OR OVERFLOWED, IF SO ERROR.
```

```
                                        ;*THE UNKNOWN THING HERE IS BIT 11 SETTING THE DISABLE F/F THAT
                                        ;*GATES WITH THE 1 MHZ FREG TO PRODUCE "A 1 MHZ CLK" L
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;*****************************************************************
013550  000004                        TST137: SCOPE
013552  012737  000100  001164              MOV     #100,$TIMES     ;;DO 100 ITERATIONS

013560  005077  165540                      CLR     @ASR            ;CLEAR CLOCK A.
013564  012777  004000  165540              MOV     #BIT11,@BSR     ;SET THE "DISABLE OSC 1 MHZ" F/F.
013572  005077  165530                      CLR     @ABR            ;CLEAR THE BUFFER + COUNT REGS.
013576  012777  000003  165520              MOV     #3,@ASR         ;START CLOCK: RATE 1 MHZ, MODE 0, GO.
013604  005000                              CLR     R0              ;SHORT DELAY. WE ALLOW THIS DELAY TO
013606  105200                      1$:     INCB    R0              ;OCCUR. IF THE 1 MHZ CLOCK IS DISABLED
013610  100376                              BPL     1$              ;NO CLOCKING OF CLOCK A WILL OCCUR.
                                                                    ;SEE SOMETHING IN THE COUNT REGISTERS
                                                                    ;OR THE OVERFLOW BIT SET.
013612  005777  165512                      TST     @ACR            ;DOES COUNT REG HAVE ANYTHING IN IT?
013616  001003                              BNE     2$              ;YES - REPORT ERROR!
013620  105777  165500                      TSTB    @ASR            ;DID OVERFLOW OCCUR?
013624  100001                              BPL     3$              ;NO - BR NEXT TEST - NO ERROR.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

013626  104012                      2$:     ERROR   12              ;ERROR - UNABLE TO DISABLE 1 MHZ CLK
                                                                    ;USING "DISABLE OSC 1 MHZ" F/F

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

013630  005077  165470          3$:     CLR     @ASR            ;CLEAR CLOCK A.
```

```
                                        .SBTTL  *
                                        .SBTTL  * PHASE 4 CLOCK B COUNT FUNCTION TESTS
                                        .SBTTL  *


                                        ;;*****************************************************************
                                        ;*TEST 140      *TEST THAT CLOCK B WILL COUNT ONCE FIRST CLOCK B COUNT
                                        ;*
                                        ;*VERY FIRST TIME COUNTING WITH CLOCK B.
                                        ;*WHAT WE'LL TRY AND DO IS COUNT IT FROM 0 TO 1.
                                        ;*WE DO HAVE A PROBLEM HERE HOWEVER. WE CAN'T READ CLOCK B AS IT
                                        ;*COUNTS AND THERE IS NO NICE WAY OF GENERATING A "CLOCK B" L PULSE.
                                        ;*SO WE'RE GOING TO HAVE TO DO A COUPLE TRICKY THINGS: (1) DISABLE
                                        ;*CLOCK A'S 1 MHZ CLOCK (WE DID THAT IN LAST TEST) SO THAT WE CAN
                                        ;*GENERATE A MAINTENANCE 1 MHZ PULSE THROUGH CLOCK A (NEVER
                                        ;*DID THAT BEFORE); (2) SET CLOCK B "FEED B TO A" BIT 05 (NEVER DID THAT
                                        ;*BEFORE EITHER) SO THAT WE CAN ROUTE "A 1 MHZ" H TO MAKE
                                        ;*"B 1 MHZ" L TO GIVE US "CLOCK B" L
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
013634  000004                        TST140: SCOPE

013636  012777  004000  165466                MOV     #BIT11,@BSR     ;CLEAR CLOCK B AND DISABLE 1MHZ OSC.
013644  005077  165464                        CLR     @BBR            ;CLEAR B'S BUFFER + COUNT REGS.
                                                                      ;SELECT "FEED B TO A", RATE 1 MHZ, GO.
013650  052777  000043  165454                BIS     #BIT5!BIT1!BIT0,@BSR  ;CLOCK A'S 1 MHZ CLOCK.
013656  052777  004000  165440                BIS     #BIT11,@ASR     ;GENERATE A MAINTENANCE 1 MHZ PULSE.
013664  005777  165446                        TST     @BCR            ;DID COUNTER COUNT?
013670  001001                              BNE     1$              ;BR IF YES TO NEXT TEST.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

013672  104014                              ERROR   14              ;ERROR - CLOCK B. FAILED TO ADVANCE
                                                                    ;COUNT REGISTER.
                                                                    ;TO SCOPE: CHECK LOAD + CLEAR INPUTS TO
                                                                    ;COUNT REG TO SEE IF NOT BEING HELD LOW.
                                                                    ;IF OK SEE IF WE'RE GETTING ANY "CLOCK B" L
                                                                    ;PULSES - WE SHOULD GET ONE EACH LOOP OF
                                                                    ;THIS SUBTEST. "CLOCK B" L COMES FROM
                                                                    ;THE B CLOCK TIMING SECTION. THE RATE IS
                                                                    ;1 MHZ. "B 1 MHZ" L IS GENERATED BY
                                                                    ;"A 1 MHZ" H + "FEED B TO A (1)" H.
                                                                    ;THE "A 1 MHZ" IS GENERATED BY "MAINT
                                                                    ;SIMUL 1 MHZ" IN CLOCK A.
        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

013674                          1$:


                                        ;;*****************************************************************
```

```
                                        ;*TEST 141      *TEST THE ABILITY IF CLOCK B TO COUNT FROM ZERO TO OVERFLOW
                                        ;*
                                        ;*LAST TEST WE FOUND OUT WE COULD ADVANCE CLOCK A'S COUNTER,
                                        ;*SO IN THIS TEST WE'RE GOING TO GO FROM 0-377 COUNTING.
                                        ;*WE'LL USE THE SAME PROCEDURE AS LAST TEST TO GENERATE "CLOCK B" L.
                                        ;*UNKNOWN IS THE ABILITY OF THE F/F'S TO PROPAGATE THEIR OVERFLOWS.
                                        ;*
                                        ;*IF IT IS DESIRED TO START THIS TEST AT A VALUE OTHER THAN 0, CHANGE
                                        ;*THE SECOND INSTR. OF THIS TEST TO A VALUE TO BE LOADED INTO THE BUFFER
                                        ;*TO THAT VALUE DESIRED.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                        ;*
                                        ;*****************************************************************
013674  000004                          TST141: SCOPE
013676  012737  000001  001164                  MOV     #1,$TIMES       ;;DO 1 ITERATION

013704  012737  000000  001124                  MOV     #0,$GDDAT       ;START THE COUNTER FROM ZERO.
                                                                        ;NOTE: A VALUE OTHER THAN ZERO MAY BE
                                                                        ;PATCHED IN HERE IN ORDER THAT A COUNT
                                                                        ;MAY BE STARTED HIGHER.

013712  005077  165406          1$:     CLR     @ASR            ;DISABLE CLOCK A.
013716  012777  004000  165406          MOV     #BIT11,@BSR     ;CLEAR CLOCK B, DISABLE 1 MHZ CLOCK A.
013724  013777  001124  165402          MOV     $GDDAT,@BBR     ;LOAD VALUE INTO COUNT + BUFFER REGS.
                                                                ;"1$" IS THE LOOP BACK POINT ON
                                                                ;"LOOP ON TEST" (SW14=1) FEATURE. NORMAL
                                                                ;LOOP BACK POINT WILL BE "2$".

013732  012777  004042  165372          MOV     #4042,@BSR      ;SELECT: "DISABLE OSC 1 MHZ"; "FEED B TO A";
                                                                ;RATE 1 MHZ.
013740  005277  165366                  INC     @BSR            ;GO. ENABL MUST BE SET AFTER "FEED B TO A"

013744  052777  004000  165352  2$:     BIS     #BIT11,@ASR     ;GENERATE A CLOCK PULSE.
                                                                ;SHOULD CAUSE THE CLOCK TO
                                                                ;INCREMENT ONCE.
013752  005237  001124                  INC     $GDDAT          ;$GDDAT IS USED TO KEEP TRACK OF THE
                                                                ;VALUE THE CLOCK SHOULD COUNT TO.

013756  017737  165354  001126          MOV     @BCR,$BDDAT     ;READ THE CLOCK'S COUNT REGISTER.

013764  123737  001126  001124          CMPB    $BDDAT,$GDDAT   ;DID THE COUNT OCCUR CORRECTLY?
013772  001402                          BEQ     3$              ;IF YES - BR "3$".

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

013774  104015                          ERROR   15              ;ERROR CLOCK B FAILED TO UPCOUNT
                                                                ;CORRECTLY - SEE COMMENTS AT SUB-TEST
                                                                ;HEADING FOR MORE DETAILS.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

013776  000403                          BR      4$
```

```
5642
5643  014000  105737  001124          3$:   TSTB   $GDDAT          ;DID WE ACHIEVE OVERFLOW YET?
5644  014004  001410                        BEQ    5$              ;
5645
5646  014006  032777  040000  165122  4$:   BIT    #BIT14,@SWR     ;LOOP ON CURRENT COUNT?
5647  014014  001753                        BEQ    2$              ;BR IF NO TO NEXT COUNT UPDATE.
5648  014016  162737  000001  001124        SUB    #1,$GDDAT       ;IF YES DECREMENT EXPECTED AND RELOAD.
5649  014024  000732                        BR     1$              ;GO TO RELOAD POINT
5650
5651  014026                          5$:                           ;END SUBTEST.
5652
5653
5654                                  ;;*****************************************************************
5655                                  ;*TEST 142      *TEST THAT CLOCK B CAN GENERATE AN OVERFLOW
5656                                  ;*
5657                                  ;*NOW WE'LL TRY AND GENERATE "B OVERFLOW" L WHICH SETS BIT 07.
5658                                  ;*WE'LL DO IT BY PRESETTING THE BUFFER TO 377 AND GENERATING A "CLOCK B" L
5659                                  ;*WE ALREADY KNOW WE CAN ADVANCE THE COUNTER, WHAT WE
5660                                  ;*WANT TO SEE IS "B OVERFLOW" L COME OVER AND DIRECT SET
5661                                  ;*"OVERFLOW FL B" F/F (BIT 07 IN CSR).
5662                                  ;*
5663                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
5664                                  ;*
5665                                  ;*
5666                                  ;*****************************************************************
5667  014026  000004                  TST142: SCOPE
5668
5669  014030  005077  165270                  CLR    @ASR            ;CLEAR CLOCK A.
5670  014034  005077  165272                  CLR    @BSR            ;CLEAR CLOCK B.
5671  014040  012777  000377  165266          MOV    #377,@BBR       ;SET COUNT AND BUFFER REGS.
5672  014046  012777  004042  165256          MOV    #4042,@BSR      ;SELECT "DISABLE OSC 1 MHZ"; "FEED B TO A";
5673                                                                 ;RATE 1 MHZ.
5674  014054  005277  165252                  INC    @BSR            ;GO. ENABL MUST BE SET AFTER "FEED B TO A"
5675  014060  052777  004000  165236          BIS    #BIT11,@ASR     ;GENERATE A CLOCK PULSE.
5676
5677  014066  105777  165240                  TSTB   @BSR            ;DID OVERFLOW FLAG SET?
5678  014072  100402                          BMI    1$              ;BR IF YES TO "1$".
5679
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5683  014074  104014                          ERROR  14              ;ERROR CLOCK B "B OVERFL FLAG" (CSR BIT7)
5684                                                                 ;FAILED TO SET ON OVERFLOW.
5685
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5689  014076  000405                          BR     2$
5690  014100  022777  000377  165230  1$:     CMP    #377,@BCR       ;WAS COUNTER RELOAD AFTER OVERFLOW
5691
5692  014106  001401                          BEQ    2$              ;BR IF YES TO NEXT TEST.
5693
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

5697  014110  104014                          ERROR  14              ;ERROR CLOCK B COUNT REG. FAILED
```

```
                                                        ;TO BE RELOADED AFTER OVERFLOW.
                ;;;$$$
```

```
                                ;;**********************************************************************
                                ;*TEST 144      *TEST THAT CLOCK B DOESN'T COUNT WHEN NO RATE IS SELECTED
                                ;;**********************************************************************
014160  000004                  TST144: SCOPE
014162  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS

014170  005077  165130                  CLR     @ASR            ;CLEAR CLOCK A.
014174  005077  165132                  CLR     @BSR            ;CLEAR CLOCK B.
014200  005077  165130                  CLR     @BBR            ;ZEROS TO BUFFER + COUNT REGS.
014204  005277  165114                  INC     @ASR            ;ENABLE COUNTER TO COUNT - RATE - 0
                                                                ;(NO RATE).
014210  005000                          CLR     R0              ;DELAY FOR ANY COUNT THAT
014212  105200                  1$:     INCB    R0              ;COULD FALSELY OCCUR.
014214  001376                          BNE     1$              ;THIS DELAY APPROX. 369 MS ON A
                                                                ;PDP 11/20.

014216  005777  165106                  TST     @ACR            ;DID ANY COUNT OCCUR?
014222  001401                          BEQ     2$              ;IF NO BR TO NEXT TEST.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

014224  104014                          ERROR   14              ;ERROR - CLOCK B COUNTED WHEN ENABLED BUT
                                                                ;NO RATE SELECTED. BETTER FIND OUT
                                                                ;WHATS GENERATING "CLOCK B" L PULSES.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

014226                  2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 145      *TEST THE ABILITY OF CLOCK B TO COUNT AT 1MHZ PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
                                        ;*IN RATE: 1MHZ PART1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
014226  000004                          TST145: SCOPE
014230  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

014236  005077  165062                          CLR     @ASR            ;/CLEAR CLOCK A.
014242  005077  165064                          CLR     @BSR            ;/CLEAR CLOCK B.
014246  005077  165062                          CLR     @BBR            ;/CLEAR THE BUFFER + COUNT REGS.
014252  012777  000003  165052                  MOV     #1!2,@BSR       ;/SELECT: RATE: 1MHZ ; GO.

014260  005000                          1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
014262  005200                                  INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
014264  001376                                  BNE     1$              ;/ON A PDP-11/20.

014266  005777  165044                          TST     @BCR            ;/DID COUNTER COUNT AT ALL?
014272  001004                                  BNE     2$              ;/BR IF YES - NEXT TEST.

014274  105777  165032                          TSTB    @BSR            ;/COUNTER MIGHT HAVE HAD TIME TO
                                                                        ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                        ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
014300  100401                                  BMI     2$              ;/BR IF SET - NEXT TEST.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

014302  104014                                  ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                        ;/AT 1MHZ RATE.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

014304                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 146      *TEST THE ABILITY OF CLOCK B TO COUNT AT 100KHZ PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
                                        ;*IN RATE: 100KHZ       PART1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
014304  000004                          TST146: SCOPE
014306  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

014314  005077  165004                          CLR     @ASR            ;/CLEAR CLOCK A.
014320  005077  165006                          CLR     @BSR            ;/CLEAR CLOCK B.
014324  005077  165004                          CLR     @BBR            ;/CLEAR THE BUFFER + COUNT REGS.
014330  012777  000005  164774                  MOV     #1!4,@BSR       ;/SELECT: RATE: 100KHZ ; GO.

014336  005000                          1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
014340  005200                                  INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
014342  001376                                  BNE     1$              ;/ON A PDP-11/20.

014344  005777  164766                          TST     @BCR            ;/DID COUNTER COUNT AT ALL?
014350  001004                                  BNE     2$              ;/BR IF YES - NEXT TEST.

014352  105777  164754                          TSTB    @BSR            ;/COUNTER MIGHT HAVE HAD TIME TO
                                                                        ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                        ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
014356  100401                                  BMI     2$              ;/BR IF SET - NEXT TEST.

                ;;;$$$
```

```
5856                                            ;/*
5857
5858                                    ;;**********************************************************************
5859                                    ;*TEST 147      *TEST THE ABILITY OF CLOCK B TO COUNT AT 1CKHZ PART 1
5860                                    ;*
5861                                    ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
5862                                    ;*IN RATE: 10KHZ        PART1
5863                                    ;*
5864                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
5865                                    ;*
5866                                    ;*
5867                                    ;;**********************************************************************
5868  014362  000004                    TST147: SCOPE
5869  014364  012737  000020  001164            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
5870
5871  014372  005077  164726            CLR     @#ASR           ;/CLEAR CLOCK A.
5872  014376  005077  164730            CLR     @#BSR           ;/CLEAR CLOCK B.
5873  014402  005077  164726            CLR     @#BBR           ;/CLEAR THE BUFFER + COUNT REGS.
5874  014406  012777  000007  164716    MOV     #1!6,@#BSR      ;/SELECT: RATE: 10KHZ ; GO.
5875
5876  014414  005000                    CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
5877  014416  005200                1$: INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
5878  014420  001376                    BNE     1$              ;/ON A PDP-11/20.
5879
5880  014422  005777  164710            TST     @#BCR           ;/DID COUNTER COUNT AT ALL?
5881  014426  001004                    BNE     2$              ;/BR IF YES - NEXT TEST.
5882
5883  014430  105777  164676            TSTB    @#BSR           ;/COUNTER MIGHT HAVE HAD TIME TO
5884                                                            ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
5885                                                            ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
5886  014434  100401                    BMI     2$              ;/BR IF SET - NEXT TEST.
5887
                ;;;$$$
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 150      *TEST THE ABILITY OF CLOCK B TO COUNT AT 1KHZ PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
                                        ;*IN RATE: 1KHZ PART1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
014440  000004                          TST150: SCOPE
014442  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

014450  005077  164650                        CLR     @ASR            ;/CLEAR CLOCK A.
014454  005077  164652                        CLR     @BSR            ;/CLEAR CLOCK B.
014460  005077  164650                        CLR     @BBR            ;/CLEAR THE BUFFER + COUNT REGS.
014464  012777  000011  164640                MOV     #1!10,@BSR      ;/SELECT: RATE: 1KHZ ; GO.

014472  005000                          1$:   CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
014474  005200                              INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
014476  001376                              BNE     1$              ;/ON A PDP-11/20.

014500  005777  164632                        TST     @BCR            ;/DID COUNTER COUNT AT ALL?
014504  001004                              BNE     2$              ;/BR IF YES - NEXT TEST.

014506  105777  164620                        TSTB    @BSR            ;/COUNTER MIGHT HAVE HAD TIME TO
                                                                    ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                    ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
014512  100401                              BMI     2$              ;/BR IF SET - NEXT TEST.

                ;;;$$$
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 151      *TEST THE ABILITY OF CLOCK B TO COUNT AT 100HZ PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
                                        ;*IN RATE: 100HZ        PART1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
014516  000004                          TST151: SCOPE
014520  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

014526  005077  164572                         CLR     @ASR            ;/CLEAR CLOCK A.
014532  005077  164574                         CLR     @BSR            ;/CLEAR CLOCK B.
014536  005077  164572                         CLR     @B8R            ;/CLEAR THE BUFFER + COUNT REGS.
014542  012777  000011  164562                 MOV     #1!11,@BSR      ;/SELECT: RATE: 100HZ ; GO.

014550  005000                         1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
014552  005200                                 INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
014554  001376                                 BNE     1$              ;/ON A PDP-11/20.

014556  005777  164554                         TST     @BCR            ;/DID COUNTER COUNT AT ALL?
014562  001004                                 BNE     2$              ;/BR IF YES - NEXT TEST.

014564  105777  164542                         TSTB    @BSR            ;/COUNTER MIGHT HAVE HAD TIME TO
                                                                       ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
014570  100401                                 BMI     2$              ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
                                                                       ;/BR IF SET - NEXT TEST.
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

014572  104014                                 ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                       ;/AT 100HZ RATE.
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

014574                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 152      *TEST THE ABILITY OF CLOCK B TO COUNT AT LINE-FREQ PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
                                        ;*IN RATE: LINE-FREQ    PART1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
014574  000004                        TST152: SCOPE
014576  012737  000020  001164                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

014604  005077  164514                       CLR     @ASR            ;/CLEAR CLOCK A.
014610  005077  164516                       CLR     @BSR            ;/CLEAR CLOCK B.
014614  005077  164514                       CLR     @BBR            ;/CLEAR THE BUFFER + COUNT REGS.
014620  012777  000017  164504               MOV     #1!16,@BSR      ;/SELECT: RATE: LINE-FREQ ; GO.

014626  005000                       1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
014630  005200                               INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
014632  001376                               BNE     1$              ;/ON A PDP-11/20.

014634  005777  164476                       TST     @BCR            ;/DID COUNTER COUNT AT ALL?
014640  001004                               BNE     2$              ;/BR IF YES - NEXT TEST.

014642  105777  164464                       TSTB    @BSR            ;/COUNTER MIGHT HAVE HAD TIME TO
                                                                     ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
014646  100401                               BMI     2$              ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
                                                                     ;/BR IF SET - NEXT TEST.
        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

014650  104014                               ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                     ;/AT LINE-FREQ RATE.
        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

014652                          2$:


                                        ;;**********************************************************************
                                        ;*TEST 153      *TEST THE "FEED B TO A" 24 BIT COUNTER FEATURE OF CLOCKS A + B
                                        ;*
                                        ;*WE'RE GOING TO TEST CLOCKS A+B AS A 24 BIT COUNTER; THAT IS;
                                        ;*WE'RE GOINT TO TAKE THE OVERFLOW FROM CLOCK B AND FEED IT INTO
                                        ;*CLOCK A.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
014652  000004                        TST153: SCOPE
```

```
MAINDEC-11-DZKWK-A     MACY11 27(732)  26-OCT-76  10:49  PAGE 135
DZKWK.CMB       T153   *TEST THE "FEED B TO A" 24 BIT COUNTER FEATURE OF CLOCKS A + B

 6038
 6039  014654  005077  164444              CLR     @ASR            ;CLEAR CLOCK A.
 6040  014660  005077  164446              CLR     @BSR            ;CLEAR CLOCK B.
 6041  014664  005077  164436              CLR     @ABR            ;CLEAR A'S BUFFER + COUNT REGISTERS.
 6042  014670  012777  000377  164436      MOV     #377,@BBR       ;PRESET B'S BUFFER + COUNT TO -1 FROM
 6043                                                              ;OVERFLOW.
 6044  014676  012777  004042  164426      MOV     #4042,@BSR      ;SELECT: "DISABLE OSC 1 MHZ"; RATE 1 MHZ;
 6045                                                              ;"FEED B TO A"
 6046  014704  005277  164422              INC     @BSR            ;SET ENABL. MUST BE SET AFTER "FEED B TO A".
 6047  014710  012777  000001  164406      MOV     #1,@ASR         ;ENABLE CLOCK A; MODE 0; RATE 0.
 6048  014716  052777  004000  164400      BIS     #BIT11,@ASR     ;SIMULATE A 1 MHZ PULSE - THIS PULSE
 6049                                                              ;WILL CLOCK CLOCK B'S COUNTER
 6050                                                              ;REGISTER. AN OVERFLOW WILL
 6051                                                              ;OCCUR - THAT OVERFLOW SHOULD
 6052                                                              ;CLOCK CLOCK A'S COUNT REGISTER.
 6053  014724  005777  164400              TST     @ACR            ;DID CLOCK A'S COUNT REG GET CLOCKED?
 6054  014730  001001                      BNE     1$              ;IF YES THEN BR NEXT TEST.
 6055

          ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS


 6059  014732  104014                      ERROR   14              ;ERROR - UNABLE TO CLOCK CLOCK A'S
 6060                                                              ;COUNT REGISTER WITH THE OVERFLOW
 6061                                                              ;FROM CLOCK B.
 6062                                                              ;THE MOST LOGICAL PLACE TO START
 6063                                                              ;WOULD BE !A CLOCK TIMING".
 6064                                                              ;"FEED B TO A (1)" H + "B OVERFLOW" H
 6065                                                              ;SHOULD BE COMING TOGETHER GOING
 6066                                                              ;INTO THE MUX - BEGING SELECTED AND
 6067                                                              ;COMING OUT OF THE MUX TO BECOME
 6068                                                              ;"CLOCK A" L.
 6069

          ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS


 6073  014734                      1$:
 6074                                      .SBTTL  *
 6075                                      .SBTTL  * PHASE 5 CLOCKS A+B INTERRUPT TESTS
 6076                                      .SBTTL  *
 6077
 6078
 6079
 6080
 6081
```

```
                        ;;*****************************************************************
                        ;*TEST 154      *TEST THAT CLOCK A WILL INTR. AND TO RIGHT VECTOR
                        ;*
                        ;*FIRST INTERRUPT TEST - CLOCK A
                        ;*
                        ;*WHEN EXECUTING THIS TEST FOR THE FIRST TIME (PASS 0) A MESSAGE
                        ;*WILL BE TYPED TO THAT EFFECT.
                        ;*IF THE PROCESSOR APPEARS TO DIE AFTER THE TYPE OUT WE CAN
                        ;*ASSUMED THAT THE CLOCK MESSED UP THE INTERRUPT SEQUENCE
                        ;*AS EXPLAINED BELOW.
                        ;*IF THE MESSAGE "TRAPPED TO LOC:XXXX FROM LOC:YYYY" IS TYPED
                        ;*WE CAN ASSUME THAT THE CLOCK ASSERTED AN INTERRUPT VECTOR
                        ;*OTHER THAN THE ONE GIVEN TO THE PROGRAM BY YOU - XXX BEING THE
                        ;*INTERRUPT VECTOR ISSUED BY THE CLOCK.
                        ;*IF THE CLOCK FAILS TO INTERRUPT THAN CHECK THE INTERRUPT
                        ;*SEQUENCE EXPLAINED BELOW.
                        ;*
                        ;*
                        ;*
                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                        ;*
                        ;*
                        ;*       ))))PDP-11-KW11K INTERRUPT SEQUENCE((((
                        ;*
                        ;* (1) CLOCK INTR FLAG GETS SET
                        ;* (2) CLOCK ISSUES A "BUS REQUEST" L
                        ;*       THIS OPTION LEAVES THE FACTORY WITH A PRIORITY CHIP FOR LEVEL '5'
                        ;* (3) PRIORITY CHIP CONVERTS "BUS REQUEST" L TO "BR6" ON UNIBUS
                        ;* (4) PROCESSOR ISSUES A "BG6 OUT" H.
                        ;* (5) PRIORITY CHIP CONVERTS THIS TO "BG OUT" H.
                        ;* (6) CLOCK ISSUES "BUS SACK" L - DROPPS "BUS REQUEST" L
                        ;* (7) PROCESSOR DROPPS "BUS BBSY" L.
                        ;* (8)*CLOCK ISSUES "BUS BBSY" L AND DROPPS "BUS SACK" L.
                        ;* (9)*CLOCK ASSERTS VECTOR ON BUS DATA LINES AND ISSUES "BUS INTR" L.
                        ;*(10) PROCESSOR ASSERTS "BUS SSYN" L.
                        ;*(11)*CLOCK DROPPS VECTOR FROM UNIBUS, DROPPS "BUS INTR" L, AND
                        ;*       "BUS BBSY" L.
                        ;*(12) PROCESSOR ASSERTS "BUS BBSY" AND TRANSFERS PROGRAM
                        ;*       CONTROL TO INTERRUPT SERVICE ROUTINE.
                        ;*
                        ;*   *    PLACES WHICH THE CLOCK COULD "HANG" THE UNIBUS.
                        ;*
                        ;;*****************************************************************
014734  000004          TST154: SCOPE


014736  005737  001206          TST     $PASS           ;/IS THIS PASS 0?
014742  001034          BNE     20$             ;/NO - DON'T TYPE OUT MESSAGE!
014744  005737  000042          TST     @#42            ;/DID WE COME HERE BY "CHAINING"?
014750  001031          BNE     20$             ;/YES - DON'T TYPE OUT MESSAGE.
014752  012737  015034  001110          MOV     #20$,$LPERR
```

```
                                                    MOV     #20$,$LPADR

                                                    TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                    BR      64$             ;;GET OVER THE ASCIZ
                                            ;;65$:  .ASCIZ  <15><12>*STARTING INTR.  TEST 154    *
                                            64$:

                                            20$:
                                                    ;*
                                                    ;*TEXT "ENTERING TEST 154 " TYPED OUT TO TELL YOU THE NEXT
                                                    ;*TEST THAT IS GOING TO BE EXECUTED. IT IS ONLY TYPED ON PASS 0.
                                                    ;*THERE IS DANGER THAT THE UNIBUS COULD GET "HUNG" WHILE
                                                    ;*EXECUTING TEST 154.
                                                    ;*
                                                    CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
                                                    CLR     @BSR            ;MAKE SURE CLOCK B IS CLEAR.

                                                    MOV     #1$,@AVECT      ;SET UP CLOCK A'S INTERRUPT VECTOR.
                                                    MOV     #340,@AVECP2    ;SET UP PSW ON INTR.
                                                                            ;SET PROCESSOR PRIORITY AT ZERO.
                                                    CLR     -(SP)
                                                    MOV     #4$,-(SP)
                                                    RTI
                                            4$:

                                                    BIS     #BIT6,@ASR
                                                    BIS     #BIT10,@BSR     ;GENERATE A CLOCK A INTR.
                                                                            ;CLOCK COULD INTR. TO WRONG VECTOR.
                                                    NOP
                                                    NOP

                    ;;;$$$
```

```
MAINDEC-11-DZKWK-A     MACY11 27(732)  26-OCT-76  10:49  PAGE 138
DZKWK.CMB       T154   *TEST THAT CLOCK A WILL INTR. AND TO RIGHT VECTOR

 6194  015140  000403                          BR      3$              ;NO INTR-GOOD!-NEXT TEST.
 6195
 6196  015142                          2$:
 6197  015142  062706  000004                  ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
 6198
               ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 6202  015146  104016                          ERROR   16              ;ERROR CLOCK A - FAILED TO CLEAR
 6203                                                                  ;"A INTR (1)" H F/F ON INTR. SP
 6204                                                                  ;A SECOND INTR WAS GENERATED.
 6205
               ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 6209  015150                          3$:
 6210
 6211
 6212                                  ;;**********************************************************************
 6213                                  ;*TEST 155      *TEST THAT CLOCK A WILL INTR WHEN CPU PSW = CLK INTR LEV -1
 6214                                  ;*
 6215                                  ;*IN THIS TEST WE'LL SEE IF CHECK A WILL INTERRUPT IF THE CPU'S
 6216                                  ;*PSW IS SET TO -1 FROM THAT OF CLOCK A. CLOCK A'S PRIORITY IS
 6217                                  ;*LEVEL 6 AS SHIPPED FROM THE FACTORY, BUT CAN BE CHANGED
 6218                                  ;*BY CHANGING THE PRIORITY JUMPER ON THE MODULE. IF THE PRIORITY LEVEL
 6219                                  ;*IS CHANGED BY INSERTING A DIFFERENT PRIORITY JUMPER YOU MUST
 6220                                  ;*PATCH A CORE LOCATION WHOSE LABEL IS "APRITY" TO THE LEVEL OF
 6221                                  ;*THE PRIORITY JUMPER INSTALLED.
 6222                                  ;*
 6223                                  ;*
 6224                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
 6225                                  ;*
 6226                                  ;;**********************************************************************
 6227  015150  000004                  TST155: SCOPE
 6228
 6229
 6230  015152  005737  001206                  TST     $PASS           ;/IS THIS PASS 0?
 6231  015156  001023                          BNE     30$             ;/IF NOT DON'T TYPE.
 6232  015160  005737  000042                  TST     @#42            ;/PROGRAM CHAINED?
 6233  015164  001020                          BNE     30$             ;/IF YES NO TYPEOUT
 6234  015166  012737  015226  001110          MOV     #30$,$LPERR
 6235  015174  012737  015226  001106          MOV     #30$,$LPADR
 6236  015202  104400  015210                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 6237  015206  000407                          BR      64$             ;;GET OVER THE ASCIZ
 6238                                  ;;65$:  .ASCIZ  #COMPLETED    #
 6239  015226                          64$:
 6240
 6241  015226                          30$:
 6242  015226  012777  015314  164104          MOV     #1$,@AVECT      ;SET INTERRUPT VECTOR ADDR.
 6243  015234  013700  001350                  MOV     APRITY,R0       ;GET INTR LEVEL.
 6244  015240  000300                          SWAB    R0              ;FIX TO GET PSW SETTING
 6245  015242  006200                          ASR     R0
 6246  015244  006200                          ASR     R0              ;PSW SETTING FOR LEVEL 6 = 300
 6247  015246  006200                          ASR     R0
 6248  015250  042700  177437                  BIC     #177437,R0      ;STRIP FOR CPU PSW SETTING AT LEVEL
 6249  015254  162700  000040                  SUB     #40,R0          ;SUB 1 LEVEL (FOR 6 = 240)
```

```
6250  015260  010046                          MOV     R0,-(SP)                ;SET CPU PSW.
6251  015262  012746  015270                  MOV     #3$,-(SP)
6252  015266  000002                          RTI
6253  015270                          3$:
6254  015270  052777  000100  164026          BIS     #BIT06,@ASR     ;ENABLE INTRS.
6255  015276  052777  002000  164026          BIS     #BIT10,@BSR     ;GENERATE A MAINTENANCE CLK A INTR.
6256
6257  015304  000240                          NOP
6258  015306  000240                          NOP
6259
              ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

6263  015310  104016                          ERROR   16              ;ERROR-CLOCK A FAILED TO INTR WHEN
6264                                                                  ;CPU'S PSW WAS SET TO -1 OF LEVEL OF
6265                                                                  ;CLOCK. EXAMINE PRIORITY JUMPER CH.P.
6266                                                                  ;PROBLEM COULD BE WRONG PRIORITY JUMPER
6267
              ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

6271  015312  000402                          BR      2$
6272
6273                                          ;*CLK INTRS. TO HERE.
6274  015314                          1$:
6275  015314  062706  000004                  ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
6276
6277  015320                          2$:
6278
6279
6280                                  ;;**************************************************************
6281                                  ;*TEST 156      *TEST THAT CLOCK A WILL NOT INTR. WHEN CPU PSW=CLK INTR LEVEL
6282                                  ;*
6283                                  ;*THIS TEST IS DESIGNED TO MAKE SURE CLOCK A DOESN'T INTERRUPT IF THE CPU'S
6284                                  ;*PSW IS SET TO THE SAME LEVEL AS THE PRIORITY OF CLOCK A. CLOCK A'S PRIORITY IS
6285                                  ;*LEVEL 6 AS SHIPPED FROM THE FACTORS, BUT CAN BE CHANGED BY CHANGING
6286                                  ;*THE PRIORITY JUMPER ON THE MODULES IF THE PRIORITY LEVEL IS CHANGED
6287                                  ;*
6288                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
6289                                  ;*
6290                                  ;*BY INSERTING A DIFFERENT PRIORITY JUMPER, YOU MUST PATCH A
6291                                  ;*CORE LOCATION WHOSE LABLE IS "APRITY" TO THE LEVEL OF THE
6292                                  ;*PRIORITY JUMPER INSTALLED.
6293                                  ;*
6294                                  ;**************************************************************
6295  015320  000004                  TST156: SCOPE
6296  015322  012737  000002  001164          MOV     #2,STIMES       ;;DO 2 ITERATIONS
6297
6298  015330  012777  015410  164002          MOV     #1$,@AVECT      ;SET INTERRUPT VECTOR ADDR.
6299  015336  013700  001350                  MOV     APRITY,R0       ;GET INTR. LEVEL
6300  015342  000300                          SWAB    R0              ;FIX TO GET PSW SETTING
6301  015344  006200                          ASR     R0
6302  015346  006200                          ASR     R0              ;PSW SETTING FOR LEVEL 6 - 300
6303  015350  006200                          ASR     R0
6304  015352  042700  177437                  BIC     #177437,R0      ;STRIP FOR CPU PSW SETTING AT LEVEL
6305  015356  010046                          MOV     R0,-(SP)                ;SET CPU PSW.
```

```
MAINDEC-11-DZKWK-A     MACY11 27(732)  26-OCT-76  10:49  PAGE 140
DZKWK.CMB       T156   *TEST THAT CLOCK A WILL NOT INTR. WHEN CPU PSW=CLK INTR LEVEL

 6306  015360  012746  015366                    MOV     #3$,-(SP)
 6307  015364  000002                            RTI
 6308  015366                          3$:
 6309  015366  052777  000100  163730            BIS     #BIT6,@ASR      ;ENABLE INTRS.
 6310  015374  052777  002000  163730            BIS     #BIT10,@BSR     ;GENERATE A MAINTENANCE CLK A INTR.
 6311  015402  000240                            NOP                     ;SHOULD HAVE INTERRUPTED IF GOING TO.
 6312  015404  000005                            RESET                   ;CLEAR CLOCK A INTR - CLK A
 6313  015406  000403                            BR      2$              ;DID NOT INTR. - GOOD.
 6314
 6315                                            ;*CLOCK INTRS TO HERE IF BAD.
 6316
 6317  015410                          1$:
 6318  015410  062706  000004                    ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
 6319
 6320
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 6324  015414  104016                            ERROR   16              ;ERROR CLOCK A INTERRUPTED WHEN
 6325                                                                    ;CPU'S PRIORITY WAS SET TO SAME
 6326                                                                    ;LEVEL AS THAT OF THE CLOCK.
 6327                                                                    ;EXAMINE PRIORITY JUMPER CHIP.
 6328                                                                    ;PROBLEM COULD BE WRONG PRIORITY JUMPER.
 6329
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 6333
 6334  015416  005077  163702          2$:       CLR     @ASR            ;CLEAR CLOCK A.
 6335
 6336
 6337                                            ;;*********************************************************************
 6338                                            ;*TEST 157      *TEST THAT ST1 WILL CAUSE CLOCK A TO INTR.
 6339                                            ;*
 6340                                            ;*WE'RE GOING TO SEE IF ST1 WILL CAUSE AN INTERRUPT.
 6341                                            ;*WE KNOW FROM PREVIOUS TESTS THAT ST1 WORKS OK-SO WHAT WE'RE
 6342                                            ;*CONCERNED WITH HERE IS IF "STP1" H AND "ST1 INTR ENB (1)" H
 6343                                            ;*WILL COME TOGETHER AND SET "A INTR (1)" H F/F.
 6344                                            ;*
 6345                                            ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
 6346                                            ;*
 6347                                            ;*
 6348                                            ;*********************************************************************
 6349  015422  000004                            TST157: SCOPE
 6350
 6351  015424  005077  163702                            CLR     @BSR            ;GENERATE A SYNC PULSE
 6352  015430  005077  163670                            CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
 6353  015434  012777  015476  163676                    MOV     #1$,@AVECT      ;SET VECTOR ADDR. FOR INTERRUPT
 6354  015442  052777  040000  163654                    BIS     #BIT14,@ASR     ;SET "ST1 INTR ENB" IN CLOCK A.
 6355  015450  052777  010000  163646                    BIS     #BIT12,@ASR     ;GENERATE A MAINTENANCE STP1.
 6356                                                                            ;CLOCK SHOULD INTERRUPT FROM HERE.
 6357                                                                            ;ALLOW INTRS.
 6358  015456  005046                                    CLR     -(SP)
 6359  015460  012746  015466                            MOV     #3$,-(SP)
 6360  015464  000002                                    RTI
 6361  015466                                  3$:
```

```
6362  015466  000240                        NOP
6363  015470  000240                        NOP
6364
6365  015472  104016                        ERROR   16              ;ERROR-CLOCK A FAILED TO GENERATE AN ST1 INTR.
6366                                                                  ;IT LOOKS AS THOUGH "STP1" + "ST1
6367                                                                  ;INTR ENB (1)" H WERE UNABLE TO TEAM
6368                                                                  ;UP TO SET "A INTR (1)" H F/F.
6369  015474  000402                        BR      2$
6370
6371                                        ;CLOCK SHOULD INTERRUPT TO HERE.
6372
6373
6374  015476                          1$:
6375  015476  062706  000004                ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
6376
6377  015502                          2$:
6378
6379
6380
6381                                        ;;*****************************************************************
6382                                        ;*TEST 160      *TEST THAT CLOCK A OVERFLOW WILL CAUSE AN INTERRUPT
6383                                        ;*
6384                                        ;*NOW A TEST TO SEE THAT CLOCK A OVERFLOW WILL CAUSE
6385                                        ;*AN INTERRUPT.
6386                                        ;*
6387                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
6388                                        ;*
6389                                        ;*
6390                                        ;;*****************************************************************
6391  015502  000004                    TST160: SCOPE
6392
6393  015504  005077  163622                CLR     @BSR            ;GENERATE A SYNC PULSE
6394  015510  005077  163610                CLR     @ASR            ;CLEAR CLOCK A.
6395  015514  012777  015564  163616        MOV     #1$,@AVECT      ;SET UP ITS INTERRUPT VECTOR.
6396  015522  012777  177777  163576        MOV     #177777,@ABR    ;SET BUFFER + COUNT REGS TO -1 FROM OVERFLOW.
6397  015530  052777  000115  163566        BIS     #BIT6!BIT3!BIT2!BIT0,@ASR  ;SET: INTERRUPT ENABLE; RATE: ST1; GO.
6398  015536  052777  010000  163560        BIS     #BIT12,@ASR     ;GENERATE A MAINTENANCE ST1.
6399                                                                  ;CLOCK SHOULD OVERFLOW CAUSING
6400                                                                  ;AN INTERRUPT.
6401
6402  015544  005046                        CLR     -(SP)           ;ALLOW INTRS.
6403  015546  012746  015554                MOV     #3$,-(SP)
6404  015552  000002                        RTI
6405  015554                          3$:
6406  015554  000240                        NOP
6407  015556  000240                        NOP
6408  015560  104016                        ERROR   16              ;ERROR-CLOCK A FAILED TO INTR. ON OVERFLOW.
6409                                                                  ;LOOKS LIKE "A OVERFLOW" L + "A INTR ENB (1)"
6410                                                                  ;UNABLE TO SET "A INTR (1)" H F/F.
6411  015562  000402                        BR      2$
6412
6413                                        ;CLOCK SHOULD INTERRUPT HERE.
6414
6415  015564                          1$:
6416  015564  062706  000004                ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
6417  015570  005077  163530            2$: CLR     @ASR            ;CLEAR CLOCK A.
```

```
                                ;;**********************************************************************
                                ;*TEST 161      *TEST THAT A CLOCK A COUNTER BUFFER CAUSES AN INTERRUPT
                                ;*
                                ;*NOW WE'LL SEE IF A COUNTER TO BUFFER TRANSACTION CAUSES.
                                ;*AN INTERRUPT.
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                ;*
                                ;*
                                ;;**********************************************************************
015574  000004                TST161: SCOPE

015576  005077  163530                CLR     @BSR            ;GENERATE A SYNC PULSE
015602  005077  163516                CLR     @ASR            ;CLEAR CLOCK A.
015606  012777  015656  163524        MOV     #1$,@AVECT      ;SET UP INTR VECTOR ADDR.
015614  012777  000340  163520        MOV     #340,@AVECP2
015622  012777  001100  163474        MOV     #1100,@ASR      ;SELECT MODE 2.
015630  052777  002000  163466        BIS     #BIT10,@ASR     ;GENERATE A MAINTENANCE ST2 PULSE.
                                                              ;AT THIS POINT A COUNTER TO BUFFER
                                                              ;TRANSACTION SHOULD OCCUR CAUSING
                                                              ;AN INTERRUPT.
015636  005046                        CLR     -(SP)           ;ALLOW INTRS.
015640  012746  015646                MOV     #3$,-(SP)
015644  000002                        RTI
015646                          3$:
015646  000240                        NOP
015650  000240                        NOP

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

015652  104016                        ERROR   16              ;ERROR CLOCK A COUNTER TO BUFFER FAILED TO
                                                              ;CAUSE AN INTERRUPT.
                                                              ;LOOK AT: "SET MODE FL" L + "A INT ENB (1)"
                                                              ;AS THEY JOIN TO SET "A INTR (1)" F/F.
                                                              ;THE ONLY WAY FOR THIS TO FAIL IS FOR
                                                              ;THE INTR BETWEEN WHERE "SET MODE FL" L
                                                              ;IS GENERATED AND THE DESTINATION TO BE
                                                              ;OPEN OR SHORTED; OR FOR A BAD
                                                              ;INPUT ON THE "7400" IC WHERE IT COMES
                                                              ;IN TO SET "A INTR (1)" H F/F.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

015654  000402                        BR      2$

                                        ;*CLOCK A SHOULD INTR TO HERE.

015656                          1$:
015656  062706  000004                ADD     #4,R6           ;ADD #4 TO THE STACK POINTER

015662  005077  163436          2$:   CLR     @ASR            ;CLEAR CLOCK A.
```

```
6474  015666  013777  001342  163444          MOV     AVECP2,@AVECT   ;ALL DONE CLOCK A INTERRUPT TESTS-SET
6475  015674  012777  104410  163440          MOV     #IOTT,@AVECP2   ;UP TO CATCH ANY ILLEGAL CLK A INTRS.
6476
6477
6478                                  ;;*****************************************************************
6479                                  ;*TEST 162      *TEST THAT CLOCK B WILL INTR. AND TO RIGHT VECTOR
6480                                  ;*
6481                                  ;*FIRST INTERRUPT TEST - CLOCK B
6482                                  ;*
6483                                  ;*WHEN EXECUTING THIS TEST FOR THE FIRST TIME (PASS 0) A MESSAGE
6484                                  ;*WILL BE TYPED TO THAT EFFECT.
6485                                  ;*IF THE PROCESSOR APPEARS TO DIE AFTER THE TYPE OUT, WE CAN
6486                                  ;*ASSUMED THAT THE CLOCK MESSED UP THE INTERRUPT SEQUENCE
6487                                  ;*AS EXPLAINED BELOW.
6488                                  ;*IF THE MESSAGE "TRAPPED TO LOC:XXXX FROM LOC:YYYY" IS TYPED
6489                                  ;*WE CAN ASSUME THAT THE CLOCK ASSERTED AN INTERRUPT VECTOR
6490                                  ;*OTHER THAN THE ONE GIVEN TO THE PROGRAM BY YOU-XXX BEING THE
6491                                  ;*INTERRUPT VECTOR ISSUED BY THE CLOCK.
6492                                  ;*IF THE CLOCK FAILES TO INTERRUPT THAN CHECK THE INTERRUPT
6493                                  ;*SEQUENCE EXPLAINED BELOW.
6494                                  ;*
6495                                  ;*
6496                                  ;*
6497                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
6498                                  ;*
6499                                  ;*
6500
6501                                  ;*      ))))PDP-11-KW11K INTERRUPT SEQUENCE((((
6502                                  ;*
6503                                  ;* (1) CLOCK INTR FLAG GETS SET
6504                                  ;* (2) CLOCK ISSUES A "BUS REQUEST" L
6505                                  ;*      THIS OPTION LEAVES THE FACTORY WITH A PRIORITY CHIP FOR LEVEL '6'
6506                                  ;* (3) PRIORITY CHIP CONVERTS "BUS REQUEST" L TO "BR6" ON UNIBUS
6507                                  ;* (4) PROCESSOR ISSUES A "BG6 OUT" H.
6508                                  ;* (5) PRIORITY CHIP CONVERTS THIS TO "BG OUT" H.
6509                                  ;* (6) CLOCK ISSUES "BUS SACK" L - DROPPS "BUS REQUEST" L
6510                                  ;* (7) PROCESSOR DROPPS "BUS BBSY" L.
6511                                  ;* (8)*CLOCK ISSUES "BUS BBSY" L AND DROPPS "BUS SACK" L.
6512                                  ;* (9)*CLOCK ASSERTS VECTOR ON BUS DATA LINES AND ISSUES "BUS INTR" L.
6513                                  ;*(10) PROCESSOR ASSERTS "BUS SSYN" L.
6514                                  ;*(11)*CLOCK DROPPS VECTOR FROM UNIBUS, DROPPS "BUS INTR" L, AND
6515                                  ;*      "BUS BBSY" L.
6516                                  ;*(12) PROCESSOR ASSERTS "BUS BBSY" AND TRANSFERS PROGRAM
6517                                  ;*      CONTROL TO INTERRUPT SERVICE ROUTINE.
6518                                  ;*
6519                                  ;*   *   PLACES WHICH THE CLOCK COULD "HANG" THE UNIBUS.
6520                                  ;*
6521
6522                                  ;;*****************************************************************
6523  015702  000004                  TST162: SCOPE
6524
6525
6526
6527  015704  005737  001206                  TST     $PASS           ;/IS THIS PASS 0?
6528  015710  001034                  BNE     20$             ;/NO - DON'T TYPE OUT MESSAGE!
6529  015712  005737  000042                  TST     @#42            ;/DID WE COME HERE BY "CHAINING"?
```

```
015716  001031                          BNE     20$             ;/YES - DON'T TYPE OUT MESSAGE.
015720  012737  016002  001110          MOV     #20$,$LPERR

015726  012737  016002  001106          MOV     #20$,$LPADR

015734  104400  015742                  TYPE    ,65$            ;;TYPE ASCIZ STRING
015740  000420                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>#STARTING INTR.  TEST 162    #
016002                          64$:


016002                          20$:
                                        ;*
                                        ;*TEXT "ENTERING TEST 162 " TYPED OUT TO TELL YOU THE NEXT
                                        ;*TEST THAT IS GOING TO BE EXECUTED. IT IS ONLY TYPED ON PASS 0.
                                        ;*THERE IS DANGER THAT THE UNIBUS COULD GET "HUNG" WHILE
                                        ;*EXECUTING TEST 162.
                                        ;*
016002  005077  163316                  CLR     @ASR            ;MAKE SURE CLOCK A IS CLEAR.
016006  005077  163320                  CLR     @BSR            ;MAKE SURE CLOCK B IS CLEAR.

016012  012777  016062  163324          MOV     #1$,@BVECT      ;SET UP CLOCK B'S INTERRUPT VECTOR.
016020  012777  000340  163320          MOV     #340,@BVECT2    ;SET UP PSW ON INTR.
016026  005046                          CLR     -(SP)           ;SET PROCESSOR PRIORITY AT ZERO.
016030  012746  016036                  MOV     #4$,-(SP)
016034  000002                          RTI
016036                          4$:

016036  052777  000100  163266          BIS     #BIT6,@BSR      ;ENABLE INTRS.
016044  052777  001000  163260          BIS     #BIT9,@BSR      ;GENERATE A CLOCK B INTR.
                                                                ;CLOCK COUNT INTR TO WRONG VECTOR
016052  000240                          NOP
016054  000240                          NOP


        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

016056  104017                          ERROR   17              ;ERROR CLOCK B FAILED TO INTERRUPT.
                                                                ;PSW = 0-GENERATED MAINTENANCE INTERRUPT
                                                                ;SEE TEST HEADING FOR LIST OF PROBLEMS

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

016060  000416                          BR      3$

                                        ;*INTRS TO HERE.

016062                          1$:
016062  062706  000004                  ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
016066  012777  016110  163250          MOV     #2$,@BVECT      ;SET UP FOR NEXT INTR. IF "B INTR (1)" H
                                                                ;WAS NOT CLEARED BY INTR.
016074  005046                          CLR     -(SP)           ;PSW=0 ALLOWING INTRS.
016076  012746  016104                  MOV     #5$,-(SP)
016102  000002                          RTI
```

```
016104                          5$:
016104  000240                          NOP             ;SHORT DELAY
016106  000403                          BR      3$      ;NO INTR-GOOD!-NEXT TEST.

016110                          2$:
016110  062706  000004                  ADD     #4,R6   ;ADD #4 TO THE STACK POINTER

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

016114  104000                          ERROR           ;ERROR CLOCK B-FAILED TO CLEAR
                                                        ;"B INTR (1)" H F/F ON INTR SO
                                                        ;A SECOND INTR WAS GENERATED

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

016116                          3$:


                                ;;**********************************************************************
                                ;*TEST 163      *TEST THAT CLOCK A WILL INTR WHEN CPU PSW CLK INTR LEV -1
                                ;*
                                ;*IN THIS TEST WE'LL SEE IF CLOCK B WILL INTERRUPT IF THE CPU'S
                                ;*PSW IS SET TO -1 FROM THAT OF CLOCK B. CLOCK B'S PRIORITY IS
                                ;*LEVEL 6 AS SHIPPED FROM THE FACTORY, BUT CAN BE CHANGED
                                ;*BY CHANGING THE PRIORITY JUMPER ON THE MODULE. IF THE PRIORITY LEVEL
                                ;*PATCH A CORE LOCATION WHOSE LABEL IS "BPRITY" TO THE LEVEL OF
                                ;*THE PRIORITY JUMPER INSTALLED.
                                ;*
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                ;*
                                ;**********************************************************************
016116  000004                  TST163: SCOPE


016120  005737  001206                  TST     $PASS           ;/IS THIS PASS 0?
016124  001023                          BNE     30$             ;/IF NOT DON'T TYPE.
016126  005737  000042                  TST     @#42            ;/PROGRAM CHAINED?
016132  001020                          BNE     30$             ;/IF YES NO TYPEOUT
016134  012737  016174  001110          MOV     #30$,$LPERR
016142  012737  016174  001106          MOV     #30$,$LPADR
016150  104400  016156                  TYPE    ,65$            ;;TYPE ASCIZ STRING
016154  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  #COMPLETED    #
016174                          64$:

016174                          30$:
016174  012777  016262  163142          MOV     #1$,@BVECT      ;SET INTERRUPT VECTOR ADDR.
016202  013700  001352                  MOV     BPRITY,R0       ;GET INTR LEVEL.
016206  000300                          SWAB    R0              ;FIX TO GET PSW SETTING
016210  006200                          ASR     R0
016212  006200                          ASR     R0              ;PSW SETTING FOR LEVEL 6-300
016214  006200                          ASR     R0
016216  042700  177437                  BIC     #177437,R0      ;STRIP FOR CPU PSW SETTING AT LEVEL
```

```
MAINDEC-11-DZKWA-A     MACY11 27(732)  26-OCT-76  10:49  PAGE 146
DZKWA.CMB       T163   *TEST THAT CLOCK A WILL INTR WHEN CPU PSW CLK INTR LEV -1

 6642  016222  162700  000040              SUB     #40,R0          ;SUB 1 LEVEL (6=240)
 6643  016226  010046                      MOV     R0,-(SP)
 6644  016230  012746  016236              MOV     #3$,-(SP)       ;SET CPU PSW.
 6645  016234  000002                      RTI
 6646  016236                        3$:
 6647  016236  052777  000100  163066      BIS     #BIT6,@BSR
 6648  016244  052777  001000  163060      BIS     #BIT9,@BSR      ;GENERATE A MAINTENANCE CLK B INTR.
 6649  016252  000240                      NOP
 6650  016254  000240                      NOP
 6651
 6652
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 6656  016256  104017                      ERROR   17              ;ERROR-CLOCK B FAILED TO INTR WHEN
 6657                                                              ;CPU'S PSW WAS SET TO -1 OF LEVEL OF
 6658                                                              ;CLOCK. EXAMINE PRIORITY JUMPER CHIP
 6659                                                              ;PROBLEM. COULD BE WRONG PRIORITY JUMPER
 6660
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 6664  016260  000402                      BR      2$
 6665
 6666                                      ;*CLK INTRS. TO HERE.
 6667
 6668  016262                        1$:
 6669  016262  062706  000004              ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
 6670
 6671  016266                        2$:
 6672  016266  005077  163040              CLR     @BSR
 6673
 6674
 6675                                ;*****************************************************************
 6676                                ;*TEST 164      *TEST THAT CLOCK B WILL NOT INTR. WHEN CPU PSW=CLK INTR LEVEL
 6677                                ;*
 6678                                ;*THIS TEST IS DESIGNED TO MAKE SURE CLOCK B DOESN'T INTERRUPT IF THE CPU'S
 6679                                ;*PSW IS SET TO THE SAME LEVEL AS THE PRIORITY OF CLOCK B. CLOCK B'S PRIORITY IS
 6680                                ;*LEVEL 6 AS SHIPPED FROM THE FACTORY, BUT CAN BE CHANGED BY CHANGING
 6681                                ;*THE PRIORITY JUMPER ON THE MODULE. IF THE PRIORITY LEVEL IS CHANGED
 6682                                ;*BY INSERTING A DIFFERENT PRIORITY JUMPER, YOU MUST PATCH A
 6683                                ;*CORE LOCATION WHOSE LABEL IS "BPRITY" TO THE LEVEL OF THE
 6684                                ;*PRIORITY JUMPER INSTALLED.
 6685                                ;*
 6686                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
 6687                                ;*
 6688                                ;*
 6689                                ;*****************************************************************
 6690  016272  000004                TST164: SCOPE
 6691  016274  012737  000002  001164       MOV     #2,$TIMES       ;;DO 2 ITERATIONS
 6692
 6693  016302  012777  016362  163034       MOV     #15,@BVECT      ;SET INTERRUPT VECTOR ADDR.
 6694  016310  013700  001352              MOV     BPRITY,R0       ;GET INTR. LEVEL
 6695  016314  000300                      SWAB    R0              ;FIX TO GET PSW SETTING
 6696  016316  006200                      ASR     R0
 6697  016320  006200                      ASR     R0              ;PSW SETTING FOR LEVEL 6-300
```

```
 6698  016322  006200                          ASR     R0
 6699  016324  042700  177437                  BIC     #177437,R0      ;STRIP FOR CPU PSW SETTING AT LEVEL
 6700  016330  010046                          MOV     R0,-(SP)
 6701  016332  012746  016340                  MOV     #3$,-(SP)       ;SET CPU PSW.
 6702  016336  000002                          RTI
 6703  016340                          3$:
 6704  016340  052777  000100  162764          BIS     #BIT6,@BSR      ;ENABLE INTRS.
 6705  016346  052777  001000  162756          BIS     #BIT9,@BSR      ;GENERATE A MAINTENANCE CLK B INTR.
 6706  016354  000240                          NOP                     ;SHOULD HAVE INTERRUPTED IF GOING TO.
 6707  016356  000005                          RESET                   ;CLEAR CLOCK B INTR-CLK B
 6708  016360  000403                          BR      2$              ;DID NOT INTR-GOOD.
 6709
 6710                                          ;*CLOCK INTRS TO HERE IF BAD.
 6711
 6712  016362                          1$:
 6713  016362  062706  000004                  ADD     #4,R6           ;ADD #4 TO THE STACK POINTER
 6714
 6715

               ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 6719  016366  104017                          ERROR   17              ;ERROR CLOCK B INTERRUPTED WHEN
 6720                                                                  ;CPU'S PRIORITY WAS SET TO SAME
 6721                                                                  ;LEVEL AS THAT OF THE CLOCK.
 6722                                                                  ;EXAMINE PRIORITY JUMPER CHIP.
 6723                                                                  ;PROBLEM COULD BE WRONG PRIORITY JUMPER
 6724

               ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 6728  016370                          2$:
 6729
 6730
 6731                                          ;;*****************************************************************
 6732                                          ;*TEST 165      *TEST THAT A CLOCK B OVERFLOW CAUSES ON INTERRUPT
 6733                                          ;*
 6734                                          ;*WE'RE GOING TO SEE NOW IF A CLOCK B OVERFLOW WILL
 6735                                          ;*CAUSE AN INTERRUPT. WE'VE ALREADY GENERATED B OVERFLOWS
 6736                                          ;*AND B INTERRUPTS SO ITS A SIMPLE MATTER OF CHECKING THE
 6737                                          ;*LOGIC IN FRONT OF "B INTR (1)" H F/F.
 6738                                          ;*
 6739                                          ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
 6740                                          ;*
 6741                                          ;*
 6742                                          ;;*****************************************************************
 6743  016370  000004                          TST165: SCOPE
 6744
 6745  016372  005077  162726                          CLR     @ASR            ;GENERATE A SYNC PULSE.
 6746  016376  005077  162730                          CLR     @BSR            ;MAKE SURE CLOCK B IS CLEAR.
 6747  016402  012777  000377  162724                  MOV     #377,@BBR       ;PRELOAD CLOCK B'S BUFFER + COUNTER REGS.
 6748  016410  012777  016454  162726                  MOV     #2$,@BVECT      ;SET UP INTERRUPT VECTOR.
 6749  016416  012777  000340  162722                  MOV     #340,@BVECT2
 6750  016424  052777  000103  162700                  BIS     #BIT6!BIT1!BIT0,@BSR  ;SET: "INTR ENB", 1 MHZ RATE, GO.
 6751  016432  005046                                  CLR     -(SP)           ;CPU PSW=0 ALLOWING INTRS.
 6752  016434  012746  016442                          MOV     #1$,-(SP)
 6753  016440  000002                                  RTI
```

```
016442                          1$:
016442  000240                          NOP
                                                        ;DELAY FOR CLOCK TO COUNT OVERFLW
                                                        ;AND INTERRUPT.
016444  000240                          NOP
016446  000240                          NOP

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

016450  104017                          ERROR   17      ;ERROR-CLOCK B FAILED TO INTERRUPT
                                                        ;ON OVERFLOW.
                                                        ;GO TO THE INTERRUPT CONTROL PAGE IN
                                                        ;YOUR PRINTS.
                                                        ;YOU SHOULD SEE "B OVERFLOW" H-ONE
                                                        ;PULSE EACH LOOP THROUGH THIS TEST.
                                                        ;"B INTR ENB (1)" SHOULD BE HIGH.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

016452  000421                          BR      6$

                                        ;CLOCK B SHOULD INTERRUPT TO HERE.

016454  012706  001100          2$:     MOV     #STACK,SP       ;RESET STACK.
016460  005077  162646          3$:     CLR     @BSR            ;CLEAR CLOCK B.
016464  012777  016510  162652          MOV     #5$,@BVECT      ;NOW WE'RE GONNA FIND OUT IF
                                                        ;"B INTR (1)" F/F GOT SET AFTER THE
                                                        ;INTR GENERATED "CLR INTR B"H
                                                        ;TO CLEAR THAT F/F. THIS CAN
                                                        ;ONLY HAPPEN IF THE CLEARING OF THE
                                                        ;STATUS REG. IN PARTICULAR,
                                                        ;CSR BIT6,DID NOT CLEAR THIS F/F.
016472  005046                          CLR     -(SP)   ;CLEAR CPU'S PSW (ALLOW INTRS.).
016474  012746  016502                  MOV     #4$,-(SP)
016500  000002                          RTI

016502  000240                  4$:     NOP             ;INTRS. FROM HERE IF ANY.
016504  000240                          NOP
016506  000403                          BR      6$      ;NO INTERRUPTS-FINE!

016510                          5$:
016510  062706  000004                  ADD     #4,R6   ;ADD #4 TO THE STACK POINTER

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

016514  104017                          ERROR   17      ;ERROR-CLOCK B INTERRUPTED ILLEGALLY!
                                                        ;CLEARING OF THE CSR,IN PARTICULAR
                                                        ;CSR BIT 6 SHOULD HAVE CLEARED THE
                                                        ;IMPENDING INTERRUPT GENERATED BY
                                                        ;HAVEING A -1 COUNT SET IN PRESET
                                                        ;BUFFER AND 1 MHZ. RATE.
```

```
                                                                        ;REVIEWING THE SEQUENCE:

                                                                        ;1. 1ST OVERFLOW CAUSED INTR. REQUEST.
                                                                        ;2. INTR OCCURED AND CLEARED "B INTR (1)"F/F.
                                                                        ;3. ANOTHER OVERFLOW OCCURED AFTER THAT
                                                                        ;   AND BEFORE THE CSR WAS CLEARED.
                                                                        ;4. CLEARING OF CSR BIT 6 FAILED TO
                                                                        ;   CLEAT "B INTR (1)" F/F,THEREBY
                                                                        ;   ANOTHER INTERRUPT STILL POSTED ON
                                                                        ;   THE UNIBUS WHEN CPU PRIORITY
                                                                        ;   WAS LOWERED.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$

016516                                  6$:
016516  013777  001346  162620                  MOV     BVECT2,@BVECT   ;NOW RESTORE INTERRUPT VECTOR
016524  012777  104410  162614                  MOV     #IOTT,@BVECT2   ;FOR ILLEGAL INTRS. THIS IS CLOCK B'S
                                                                        ;LAST INTERRUPT TEST.

                                        .SBTTL  *
                                        .SBTTL  * PHASE 6 CLOCK A+B ADVANCE TESTING
                                        .SBTTL  *
```

```
6834                                        ;/*
6835
6836                                    ;;*************************************************************
6837                                    ;*TEST 166      *TEST THAT THE TRAILING EDGE OF STP1 WILL INCR. COUNTER
6838                                    ;*
6839                                    ;*IN THIS TEST WE'LL SEE IF WE CATCH THE FIRST COUNT
6840                                    ;*AFTER STP1 COMES IN AND SETS THE ENABLE F/F ('ST1 ENB COUNTER'
6841                                    ;*SET).
6842                                    ;*WHAT WE SHOULD SEE IS THE LEADING EDGE OF ST1 COME
6843                                    ;*IN AND SET THE ENB F/F AND THE TRAILING EDGE TRIGGER
6844                                    ;*A 'CLOCK A' PULSE TO INCREMENT THE COUNTER.
6845                                    ;*WE KNOW FROM A PREVIOUS TEST THAT AN STP1 WILL
6846                                    ;*COME IN AND SET 'ENABL CNTR A' F/F AND THAT THE
6847                                    ;*COUNTER WILL INCREMENT, SO WHATS HAPPENING IS WE'RE ACCUALLY
6848                                    ;*LOOKING AT THE TRAILING EDGE OF STP1 TO SEE IF ITS DOING
6849                                    ;*THE INCREMENTING
6850                                    ;*
6851                                    ;*
6852                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
6853                                    ;*
6854                                    ;*************************************************************
6855  016532  000004                    TST166: SCOPE
6856
6857  016534  012777  020014  162562            MOV     #BIT13!14,@ASR  ;CLR CLK A, SET 'ST1 ENB COUNTER'.;RATE:STP1.
6858  016542  005077  162560                    CLR     @ABR            ;CLR COUNT + BUFFER REGS.
6859  016546  052777  010000  162550            BIS     #BIT12,@ASR     ;GENERATE A MAINTENANCE ST1.
6860                                                                    ;THE LEADING EDGE SHOULD CAUSE
6861                                                                    ;'.ENABL CNTR A' F/F TO SET (CSR BIT 00).
6862                                                                    ;THE TRAILING EDGE SHOULD CLOCK
6863                                                                    ;THE COUNTER ONCE.
6864
6865  016554  012737  000001  001124            MOV     #1,$GDDAT       ;SET S/B FOR ERROR TYPEOUT IF NEEDED.
6866  016562  017737  162542  001126            MOV     @ACR,$BDDAT     ;READ THE COUNT REGISTER.
6867  016570  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;DID THE COUNT REG COUNT ONCE?
6868  016576  001401                            BEQ     1$              ;BR IF YES TO NEXT TEST.
6869
6870
                  ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS


6874
6875  016600  104012                            ERROR   12              ;ST1 FAILED TO COUNT
6876                                                                    ;CLOCK A'S COUNT REG. AFTER SETTING
6877                                                                    ;'ENABL CNTR A' - SEE TEST HEADING
6878                                                                    ;COMMENTS
6879
6880
                  ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS


6884
6885  016602                            1$:
6886
6887
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 167      *TEST CLOCK A'S 100KHZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 100KHZ DIVIDER WILL DIVIDE 1MHZ
                                        ;*BY 10 TO GIVE US A 100KHZ CLK L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
                                        ;*GENERATE 10. '1MHZ CLK' H PULSES WHICH GIVES US 10 1MHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100KHZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 1MHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 100KHZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************************
016602  000004                          TST167: SCOPE
016604  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

016612  005077  162514                          CLR     @BSR            ;/CLEAR CLOCK B.
016616  005077  162502                          CLR     @ASR            ;/CLEAR CLOCK A.
016622  005077  162500                          CLR     @ABR            ;/CLEAR A'S BUFFER + COUNT REGS.
016626  012777  004000  162476                  MOV     #BIT11,@BSR     ;/DISABLE THE 1MHZ OSC.
016634  052777  000405  162462                  BIS     #401!4,@ASR     ;/ENABLE CNTR, RATE: 100KHZ ;MODE.

016642  012700  177766                          MOV     #-10.,R0        ;/SET TO GENERATE 10. 1MHZ PULSES

016646  052777  004000  162450  1$:             BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
016654  005777  162450                          TST     @ACR            ;/HAS COUNTER ADVANCED ANY?
016660  001002                                  BNE     10$             ;/IF SO EXIT THIS LOOP.
                                                                        ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                                        ;/OSC.,THE DIVIDER COULD HAVE
                                                                        ;/AND COUNT LEFT IN IT.
                                                                        ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
016662  005200                                  INC     R0              ;/DONE 10. 1MHZ PULSES?
016664  001370                                  BNE     1$              ;/IF NOT - DO ANOTHER.

016666  012737  000001  001124  10$:            MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

016674  017737  162430  001126                  MOV     @ACR,$BDDAT     ;/READ THE COUNTER.

016702  023737  001126  001124                  CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
016710  001402                                  BEQ     2$              ;/IF YES - NEXT CHECK.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$

016712  104012                                  ERROR   12              ;/ERROR - CLOCK A - 100KHZ - PULSE
                                                                        ;/NOT GENERATED WHEN 10 1MHZ PULSES

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$

016714  000417                                  BR      4$
```

```
 6944  016716  012700  000011          2$:     MOV     #9.,R0  ;/GET THE NUMBER OF '1 MHZ' H PULSES
 6945
 6946  016722  052777  004000  162374  3$:     BIS     #BIT11,@ASR     ;/GENERATE 9 1MHZ PULSES
 6947  016730  005300                          DEC     R0              ;/INORDER TO CHECK TO SEE THAT
 6948  016732  001373                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 100KHZ PULSE.
 6949
 6950  016734  017737  162370  001126          MOV     @ACR,$BDDAT     ;/READ THE COUNTER
 6951  016742  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100KHZ PULSE GENERATED?
 6952  016750  001401                          BEQ     4$              ;/NO-GOTO NEXT TEST!
 6953
 6954

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 6958  016752  104012                          ERROR   12              ;/ERROR CLOCK A WE SEEM TO HAVE
 6959                                                                  ;/GENERATED A SECOND 100KHZ PULSE
 6960                                                                  ;/ON ONLY 9 1MHZ PULSES.
 6961

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 6965  016754                          4$:
```

```
                                        ;/*
                                        ;;*****************************************************
                                        ;*TEST 170      *TEST CLOCK A'S 10KHZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 10KHZ DIVIDER WILL DIVIDE 100KHZ
                                        ;*BY 10 TO GIVE US A 10KHZ CLK L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
                                        ;*GENERATE 100. '1MHZ CLK' H PULSES WHICH GIVES US 10 100KHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 10KHZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 100KHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 10KHZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************
016754  000004                        TST170: SCOPE
016756  012737  000020  001164                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

016764  005077  162342                        CLR     @BSR            ;/CLEAR CLOCK B.
016770  005077  162330                        CLR     @ASR            ;/CLEAR CLOCK A.
016774  005077  162326                        CLR     @ABR            ;/CLEAR A'S BUFFER + COUNT REGS.
017000  012777  004000  162324                MOV     #BIT11,@BSR     ;/DISABLE THE 1MHZ OSC.
017006  052777  000407  162310                BIS     #401!6,@ASR     ;/ENABLE CNTR, RATE: 10KHZ ;MODE.

017014  012700  177634                        MOV     #-100.,R0       ;/SET TO GENERATE 100. 1MHZ PULSES

017020  052777  004000  162276  1$:   BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
017026  005777  162276                TST     @ACR            ;/HAS COUNTER ADVANCED ANY?
017032  001002                        BNE     10$             ;/IF SO EXIT THIS LOOP.
                                                              ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                              ;/OSC., THE DIVIDER COULD HAVE
                                                              ;/AND COUNT LEFT IN IT.
                                                              ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
017034  005200                        INC     R0              ;/DONE 100. 1MHZ PULSES?
017036  001370                        BNE     1$              ;/IF NOT - DO ANOTHER.

017040  012737  000001  001124  10$:   MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

017046  017737  162256  001126        MOV     @ACR,$BDDAT     ;/READ THE COUNTER.

017054  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
017062  001402                        BEQ     2$              ;/IF YES - NEXT CHECK.

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

017064  104012                        ERROR   12              ;/ERROR - CLOCK A - 10KHZ - PULSE
                                                              ;/NOT GENERATED WHEN 10 100KHZ PULSES

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

017066  000417                        BR      4$
```

```
7022  017070  012700  000143          2$:   MOV   #99.,R0 ;/GET THE NUMBER OF '1 MHZ' H PULSES
7023
7024  017074  052777  004000  162222  3$:   BIS   #BIT11,@ASR     ;/GENERATE 9 100KHZ PULSES
7025  017102  005300                        DEC   R0              ;/INORDER TO CHECK TO SEE THAT
7026  017104  001373                        BNE   3$              ;/WE DON'T GENERATE ANOTHER 10KHZ PULSE.
7027
7028  017106  017737  162216  001126        MOV   @ACR,$BDDAT     ;/READ THE COUNTER
7029  017114  023737  001126  001124        CMP   $BDDAT,$GDDAT   ;/WAS ANOTHER 10KHZ PULSE GENERATED?
7030  017122  001401                        BEQ   4$              ;/NO-GOTO NEXT TEST!
7031
7032

              ;;;$$$
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 171      *TEST CLOCK A'S 1KHZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 1KHZ DIVIDER WILL DIVIDE 10KHZ
                                        ;*BY 10 TO GIVE US A 1KHZ CLK L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
                                        ;*GENERATE 1000. '1MHZ CLK' H PULSES WHICH GIVES US 10 10KHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 1KHZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 10KHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 1KHZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************************
017126  000004                          TST171: SCOPE
017130  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

017136  005077  162170                          CLR     @BSR            ;/CLEAR CLOCK B.
017142  005077  162156                          CLR     @ASR            ;/CLEAR CLOCK A.
017146  005077  162154                          CLR     @ABR            ;/CLEAR A'S BUFFER + COUNT REGS.
017152  012777  004000  162152                  MOV     #BIT11,@BSR     ;/DISABLE THE 1MHZ OSC.
017160  052777  000411  162136                  BIS     #401!10,@ASR    ;/ENABLE CNTR, RATE: 1KHZ ;MODE.

017166  012700  176030                          MOV     #-1000.,R0      ;/SET TO GENERATE 1000. 1MHZ PULSES

017172  052777  004000  162124  1$:             BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
017200  005777  162124                          TST     @ACR            ;/HAS COUNTER ADVANCED ANY?
017204  001002                                  BNE     10$             ;/IF SO EXIT THIS LOOP.
                                                                        ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                                        ;/OSC.,THE DIVIDER COULD HAVE
                                                                        ;/AND COUNT LEFT IN IT.
                                                                        ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
017206  005200                                  INC     R0              ;/DONE 1000. 1MHZ PULSES?
017210  001370                                  BNE     1$              ;/IF NOT - DO ANOTHER.

017212  012737  000001  001124  10$:            MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

017220  017737  162104  001126                  MOV     @ACR,$BDDAT     ;/READ THE COUNTER.

017226  023737  001126  001124                  CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
017234  001402                                  BEQ     2$              ;/IF YES - NEXT CHECK.

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

017236  104012                                  ERROR   12              ;/ERROR - CLOCK A - 1KHZ - PULSE
                                                                        ;/NOT GENERATED WHEN 10 10KHZ PULSES

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

017240  000417                                  BR      4$
```

```
 7100  017242  012700  001747          2$:    MOV    #999.,R0          ;/GET THE NUMBER OF '1 MHZ' H PULSES
 7101
 7102  017246  052777  004000  162050  3$:    BIS    #BIT11,@ASR       ;/GENERATE 9 10KHZ PULSES
 7103  017254  005300                        DEC    R0                ;/INORDER TO CHECK TO SEE THAT
 7104  017256  001373                        BNE    3$                ;/WE DON'T GENERATE ANOTHER 1KHZ PULSE.
 7105
 7106  017260  017737  162044  001126        MOV    @ACR,$BDDAT       ;/READ THE COUNTER
 7107  017266  023737  001126  001124        CMP    $BDDAT,$GDDAT     ;/WAS ANOTHER 1KHZ PULSE GENERATED?
 7108  017274  001401                        BEQ    4$                ;/NO-GOTO NEXT TEST!
 7109
 7110

               ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 7114  017276  104012                        ERROR  12                ;/ERROR CLOCK A WE SEEM TO HAVE
 7115                                                                 ;/GENERATED A SECOND 1KHZ PULSE
 7116                                                                 ;/ON ONLY 9 10KHZ PULSES.
 7117

               ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 7121  017300                          4$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 172      *TEST CLOCK A'S 100HZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 100HZ DIVIDER WILL DIVIDE 1KHZ
                                        ;*BY 10 TO GIVE US A 100HZ CLK L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
                                        ;*GENERATE 10000. '1MHZ CLK' H PULSES WHICH GIVES US 10 1KHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100HZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 1KHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 100HZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*****************************************************************
017300  000004                          TST172: SCOPE
017302  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

017310  005077  162016                          CLR     @BSR            ;/CLEAR CLOCK B.
017314  005077  162004                          CLR     @ASR            ;/CLEAR CLOCK A.
017320  005077  162002                          CLR     @ABR            ;/CLEAR A'S BUFFER + COUNT REGS.
017324  012777  004000  162000                  MOV     #BIT11,@BSR     ;/DISABLE THE 1MHZ OSC.
017332  052777  000413  161764                  BIS     #401!12,@ASR    ;/ENABLE CNTR, RATE: 100HZ ;MODE.

017340  012700  154360                          MOV     #-10000.,R0     ;/SET TO GENERATE 10000. 1MHZ PULSES

017344  052777  004000  161752  1$:     BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
017352  005777  161752                  TST     @ACR            ;/HAS COUNTER ADVANCED ANY?
017356  001002                          BNE     10$             ;/IF SO EXIT THIS LOOP.
                                                                ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                                ;/OSC. THE DIVIDER COULD HAVE
                                                                ;/AND COUNT LEFT IN IT.
                                                                ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
017360  005200                          INC     R0              ;/DONE 10000. 1MHZ PULSES?
017362  001370                          BNE     1$              ;/IF NOT - DO ANOTHER.

017364  012737  000001  001124  10$:     MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

017372  017737  161732  001126          MOV     @ACR,$BDDAT     ;/READ THE COUNTER.

017400  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
017406  001402                          BEQ     2$              ;/IF YES - NEXT CHECK.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

017410  104012                          ERROR   12              ;/ERROR - CLOCK A - 100HZ - PULSE
                                                                ;/NOT GENERATED WHEN 10 1KHZ PULSES

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

017412  000417                          BR      4$
```

```
MAINDEC-11-DZKWK-A     MACY11 27(732)  26-OCT-76  10:49  PAGE 158
DZKWK.CMB       T172   *TEST CLOCK A'S 100HZ DIVIDER

 7178  017414  012700  023417          2$:     MOV     #9999.,R0       ;/GET THE NUMBER OF '1 MHZ' H PULSES
 7179
 7180  017420  052777  004000  161676  3$:     BIS     #BIT11,@ASR     ;/GENERATE 9 1KHZ PULSES
 7181  017426  005300                          DEC     R0              ;/INORDER TO CHECK TO SEE THAT
 7182  017430  001373                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 100HZ PULSE.
 7183
 7184  017432  017737  161672  001126          MOV     @ACR,$BDDAT     ;/READ THE COUNTER
 7185  017440  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100HZ PULSE GENERATED?
 7186  017446  001401                          BEQ     4$              ;/NO-GOTO NEXT TEST!
 7187
 7188

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

 7192  017450  104012                          ERROR   12              ;/ERROR CLOCK A WE SEEM TO HAVE
 7193                                                                  ;/GENERATED A SECOND 100HZ PULSE
 7194                                                                  ;/ON ONLY 9 1KHZ PULSES.
 7195

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS

 7199  017452                          4$:
 7200
 7201
 7202
 7203
 7204
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 173      *TEST CLOCK A'S REPEATIBILITY AT 1MHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 1MHZ REPEATABILITY OF
                                        ;*CLOCK A (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +-2.).
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;;*****************************************************************
017452  000004                          TST173: SCOPE
017454  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

017462  005077  161636                          CLR     @ASR            ;/CLEAR CLOCK A.
017466  005077  161640                          CLR     @BSR            ;/CLEAR CLOCK B.
017472  005077  161630                          CLR     @ABR            ;/CLEAR CLOCK A'S BUFFER REG.
017476  012777  000002  161620                  MOV     #2,@ASR         ;/SET RATE: 1MHZ.
017504  012700  000036                          MOV     #30.,R0 ;/SET THE DELAY.

017510  005277  161610                          INC     @ASR            ;/ENABLE THE COUNTER TO COUNT

017514  005300                          1$:     DEC     R0              ;/DELAY.
017516  001376                                  BNE     1$

017520  005077  161600                          CLR     @ASR            ;/STOP THE CLOCK.
017524  017737  161600  001124                  MOV     @ACR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

017532  005077  161570                          CLR     @ABR            ;/RELOAD THE BUF. REG.
017536  012777  000002  161560                  MOV     #2,@ASR         ;/SET RATE: 1MHZ.
017544  012700  000036                          MOV     #30.,R0 ;/SET THE DELAY.

017550  005277  161550                          INC     @ASR            ;/ENABLE THE COUNTER TO COUNT.

017554  005300                          2$:     DEC     R0              ;/DELAY (SAME AS AT "1$").
017556  001376                                  BNE     2$

017560  005077  161540                          CLR     @ASR            ;/STOP THE CLOCK!
017564  017737  161540  001126                  MOV     @ACR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

017572  013700  001124                          MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
017576  163700  001126                          SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                        ;/IN ORDER TO FIND OUT WHAT THE
                                                                        ;/VARIENCE WAS.
017602  100001                                  BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
017604  005400                                  NEG     R0              ;/DO IF SUB WAS A NEG RESULT,
                                                                        ;/MAKE IT POSITIVE.
017606                                  3$:
017606  020027  000002                          CMP     R0,#2           ;/DID THE TWO COUNTS UP VARY
                                                                        ;/MORE THAN 2?

017612  003402                                  BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

                                        ERROR   20              ;/ERROR - CLOCK A - 1MHZ REPEATABILITY
                                                                ;/VARIENCE GREATER THAN 2.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

017616  000416                    BR      6$
017620  012737  000001  001162  4$: MOV     #1,$TMP0        ;/SET MINIMUM COUNT EXPECTED.
017626  023737  001162  001124      CMP     $TMP0,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
                                                            ;/COUNT EXPECTED?
017634  003402                    BLE     5$              ;/BR IF YES - NEXT CHECK

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

017636  104021                    ERROR   21              ;/ERROR CLOCK A 1MHZ, MINIMUM
                                                            ;/COUNT NOT REACHED DURING DELAY,
                                                            ;/1ST COUNT PERIOD.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

017640  000405                    BR      6$
017642  023737  001162  001126  5$: CMP     $TMP0,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
                                                            ;/COUNT EXPECTED?
017650  003401                    BLE     6$              ;/BR IF YES - NEXT TEST.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

017652  104022                    ERROR   22              ;/ERROR CLOCK A 1MHZ, MINIMUM
                                                            ;/COUNT NOT REACHED DURING DELAY,
                                                            ;/SECOND COUNT PERIOD.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS


017654                      6$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 174      *TEST CLOCK A'S REPEATIBILITY AT 100KHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 100KHZ REPEATABILITY OF
                                        ;*CLOCK A (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +-2.).
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;;**********************************************************************
017654  000004                        TST174: SCOPE
017656  012737  000020  001164                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

017664  005077  161434                        CLR     @ASR            ;/CLEAR CLOCK A.
017670  005077  161436                        CLR     @BSR            ;/CLEAR CLOCK B.
017674  005077  161426                        CLR     @ABR            ;/CLEAR CLOCK A'S BUFFER REG.
017700  012777  000004  161416                MOV     #4,@ASR         ;/SET RATE: 100KHZ.
017706  012700  000454                        MOV     #300.,R0        ;/SET THE DELAY.

017712  005277  161406                        INC     @ASR            ;/ENABLE THE COUNTER TO COUNT

017716  005300                        1$:     DEC     R0              ;/DELAY.
017720  001376                                BNE     1$

017722  005077  161376                        CLR     @ASR            ;/STOP THE CLOCK.
017726  017737  161376  001124                MOV     @ACR,$GDDAT     ;/REAL THE COUNTER, STORE IN "$GDDAT".

017734  005077  161366                        CLR     @ABR            ;/RELOAD THE BUF. REG.
017740  012777  000004  161356                MOV     #4,@ASR         ;/SET RATE: 100KHZ.
017746  012700  000454                        MOV     #300.,R0        ;/SET THE DELAY.

017752  005277  161346                        INC     @ASR            ;/ENABLE THE COUNTER TO COUNT.

017756  005300                        2$:     DEC     R0              ;/DELAY (SAME AS AT "1$").
017760  001376                                BNE     2$

017762  005077  161336                        CLR     @ASR            ;/STOP THE CLOCK!
017766  017737  161336  001126                MOV     @ACR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

017774  013700  001124                        MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
020000  163700  001126                        SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                      ;/IN ORDER TO FIND OUT WHAT THE
                                                                      ;/VARIENCE WAS.
020004  100001                                BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
020006  005400                                NEG     R0              ;/DO IF SUB WAS A NEG RESULT,
                                                                      ;/MAKE IT POSITIVE.
020010                                  3$:
020010  020027  000002                        CMP     R0,#2           ;/DID THE TWO COUNTS UP VARY
                                                                      ;/MORE THAN 2?

020014  003402                                BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

020016  104020                          ERROR   20              ;/ERROR - CLOCK A - 100KHZ REPEATABILITY
                                                                ;/VARIENCE GREATER THAN 2.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

020020  000416                          BR      6$

020022  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.

020030  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
                                                                ;/COUNT EXPECTED?

020036  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

020040  104021                          ERROR   21              ;/ERROR CLOCK A 100KHZ, MINIMUM
                                                                ;/COUNT NOT REACHED DURING DELAY,
                                                                ;/1ST COUNT PERIOD.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

020042  000405                          BR      6$

020044  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
                                                                ;/COUNT EXPECTED?
020052  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

020054  104022                          ERROR   22              ;/ERROR CLOCK A 100KHZ, MINIMUM
                                                                ;/COUNT NOT REACHED DURING DELAY,
                                                                ;/SECOND COUNT PERIOD.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS


020056                          6$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 175      *TEST CLOCK A'S REPEATIBILITY AT 10KHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 10KHZ REPEATABILITY OF
                                        ;*CLOCK A (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +-2.).
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;;*****************************************************************
020056  000004                          TST175: SCOPE
020060  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

020066  005077  161232                         CLR     @ASR            ;/CLEAR CLOCK A.
020072  005077  161234                         CLR     @BSR            ;/CLEAR CLOCK B.
020076  005077  161224                         CLR     @ABR            ;/CLEAR CLOCK A'S BUFFER REG.
020102  012777  000006  161214                 MOV     #6,@ASR         ;/SET RATE: 10KHZ.
020110  012700  005670                         MOV     #3000.,R0       ;/SET THE DELAY.

020114  005277  161204                         INC     @ASR            ;/ENABLE THE COUNTER TO COUNT

020120  005300                          1$:    DEC     R0              ;/DELAY.
020122  001376                                 BNE     1$

020124  005077  161174                         CLR     @ASR            ;/STOP THE CLOCK.
020130  017737  161174  001124                 MOV     @ACR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

020136  005077  161164                         CLR     @ABR            ;/RELOAD THE BUF. REG.
020142  012777  000006  161154                 MOV     #6,@ASR         ;/SET RATE: 10KHZ.
020150  012700  005670                         MOV     #3000.,R0       ;/SET THE DELAY.

020154  005277  161144                         INC     @ASR            ;/ENABLE THE COUNTER TO COUNT.

020160  005300                          2$:    DEC     R0              ;/DELAY (SAME AS AT "1$").
020162  001376                                 BNE     2$

020164  005077  161134                         CLR     @ASR            ;/STOP THE CLOCK!
020170  017737  161134  001126                 MOV     @ACR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

020176  013700  001124                         MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
020202  163700  001126                         SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                       ;/IN ORDER TO FIND OUT WHAT THE
                                                                       ;/VARIENCE WAS.
020206  100001                                 BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
020210  005400                                 NEG     R0              ;/DO IF SUB WAS A NEG RESULT,
                                                                       ;/MAKE IT POSITIVE.
020212                          3$:
020212  020027  000002                         CMP     R0,#2           ;/DID THE TWO COUNTS UP VARY
                                                                       ;/MORE THAN 2?

020216  003402                                 BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7475  020220  104020                          ERROR   20              ;/ERROR - CLOCK A - 10KHZ REPEATABILITY
7476                                                                 ;/VARIENCE GREATER THAN 2.
7477
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7481  020222  000416                          BR      6$
7482
7483  020224  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.
7484
7485  020232  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
7486                                                                 ;/COUNT EXPECTED?
7487
7488  020240  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
7489
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7493  020242  104021                          ERROR   21              ;/ERROR CLOCK A 10KHZ, MINIMUM
7494                                                                 ;/COUNT NOT REACHED DURING DELAY,
7495                                                                 ;/1ST COUNT PERIOD.
7496
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7500  020244  000405                          BR      6$
7501
7502  020246  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
7503                                                                 ;/COUNT EXPECTED?
7504  020254  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
7505
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7509  020256  104022                          ERROR   22              ;/ERROR CLOCK A 10KHZ, MINIMUM
7510                                                                 ;/COUNT NOT REACHED DURING DELAY,
7511                                                                 ;/SECOND COUNT PERIOD.
7512
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7516
7517  020260                          6$:
7518
7519
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 176      *TEST CLOCK A'S REPEATIBILITY AT 1KHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 1KHZ REPEATABILITY OF
                                        ;*CLOCK A (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +-2.).
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;;**********************************************************************
020260  000004                        TST176: SCOPE
020262  012737  000020  001164                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

020270  005077  161030                       CLR     @ASR            ;/CLEAR CLOCK A.
020274  005077  161032                       CLR     @BSR            ;/CLEAR CLOCK B.
020300  005077  161022                       CLR     @ABR            ;/CLEAR CLOCK A'S BUFFER REG.
020304  012777  000010  161012               MOV     #10,@ASR                ;/SET RATE: 1KHZ.
020312  012700  072460                       MOV     #30000.,R0      ;/SET THE DELAY.

020316  005277  161002                       INC     @ASR            ;/ENABLE THE COUNTER TO COUNT

020322  005300                        1$:    DEC     R0              ;/DELAY.
020324  001376                               BNE     1$

020326  005077  160772                       CLR     @ASR            ;/STOP THE CLOCK.
020332  017737  160772  001124               MOV     @ACR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

020340  005077  160762                       CLR     @ABR            ;/RELOAD THE BUF. REG.
020344  012777  000010  160752               MOV     #10,@ASR                ;/SET RATE: 1KHZ.
020352  012700  072460                       MOV     #30000.,R0      ;/SET THE DELAY.

020356  005277  160742                       INC     @ASR            ;/ENABLE THE COUNTER TO COUNT.

020362  005300                        2$:    DEC     R0              ;/DELAY (SAME AS AT "1$").
020364  001376                               BNE     2$

020366  005077  160732                       CLR     @ASR            ;/STOP THE CLOCK!
020372  017737  160732  001126               MOV     @ACR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

020400  013700  001124                       MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
020404  163700  001126                       SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                     ;/IN ORDER TO FIND OUT WHAT THE
                                                                     ;/VARIENCE WAS.
020410  100001                               BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
020412  005400                               NEG     R0              ;/DO IF SUB WAS A NEG RESULT,
                                                                     ;/MAKE IT POSITIVE.
020414                                  3$:
020414  020027  000002                       CMP     R0,#2           ;/DID THE TWO COUNTS UP VARY
                                                                     ;/MORE THAN 2?

020420  003402                               BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

020422  104020                          ERROR   20              ;/ERROR - CLOCK A - 1KHZ REPEATABILITY
                                                                ;/VARIENCE GREATER THAN 2.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

020424  000416                          BR      6$

020426  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.

020434  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
                                                                ;/COUNT EXPECTED?

020442  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

020444  104021                          ERROR   21              ;/ERROR CLOCK A 1KHZ, MINIMUM
                                                                ;/COUNT NOT REACHED DURING DELAY,
                                                                ;/1ST COUNT PERIOD.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

020446  000405                          BR      6$

020450  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
                                                                ;/COUNT EXPECTED?
020456  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

020460  104022                          ERROR   22              ;/ERROR CLOCK A 1KHZ, MINIMUM
                                                                ;/COUNT NOT REACHED DURING DELAY,
                                                                ;/SECOND COUNT PERIOD.

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$


020462                  6$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 177      *TEST CLOCK A'S REPEATIBILITY AT 100HZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 100HZ REPEATABILITY OF
                                        ;*CLOCK A (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +-2.).
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                        ;;**********************************************************************
020462  000004                    TST177: SCOPE
020464  012737  000020  001164            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
020472  005077  160626                    CLR     @ASR            ;/CLEAR CLOCK A.
020476  005077  160630                    CLR     @BSR            ;/CLEAR CLOCK B.
020502  005077  160620                    CLR     @ABR            ;/CLEAR CLOCK A'S BUFFER REG.
020506  012777  000012  160610            MOV     #12,@ASR                ;/SET RATE: 100HZ.
020514  012700  177777                    MOV     #-1,R0  ;/SET THE DELAY.

020520  005277  160600                    INC     @ASR            ;/ENABLE THE COUNTER TO COUNT

020524  005300                    1$:     DEC     R0              ;/DELAY.
020526  001376                            BNE     1$

020530  005077  160570                    CLR     @ASR            ;/STOP THE CLOCK.
020534  017737  160570  001124            MOV     @ACR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

020542  005077  160560                    CLR     @ABR            ;/RELOAD THE BUF. REG.
020546  012777  000012  160550            MOV     #12,@ASR                ;/SET RATE: 100HZ.
020554  012700  177777                    MOV     #-1,R0  ;/SET THE DELAY.

020560  005277  160540                    INC     @ASR            ;/ENABLE THE COUNTER TO COUNT.

020564  005300                    2$:     DEC     R0              ;/DELAY (SAME AS AT "1$").
020566  001376                            BNE     2$

020570  005077  160530                    CLR     @ASR            ;/STOP THE CLOCK!
020574  017737  160530  001126            MOV     @ACR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

020602  013700  001124                    MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
020606  163700  001126                    SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                ;/IN ORDER TO FIND OUT WHAT THE
                                                                ;/VARIENCE WAS.
020612  100001                            BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
020614  005400                            NEG     R0              ;/DO IF SUB WAS A NEG RESULT,
                                                                ;/MAKE IT POSITIVE.
020616                            3$:
020616  020027  000002                    CMP     R0,#2           ;/DID THE TWO COUNTS UP VARY
                                                                ;/MORE THAN 2?

020622  003402                            BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7685  020624  104020                          ERROR   20              ;/ERROR - CLOCK A - 100HZ REPEATABILITY
7686                                                                  ;/VARIENCE GREATER THAN 2.
7687
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7691  020626  000416                          BR      6$
7692
7693  020630  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.
7694
7695  020636  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
7696                                                                  ;/COUNT EXPECTED?
7697
7698  020644  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
7699
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7703  020646  104021                          ERROR   21              ;/ERROR CLOCK A 100HZ, MINIMUM
7704                                                                  ;/COUNT NOT REACHED DURING DELAY,
7705                                                                  ;/1ST COUNT PERIOD.
7706
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7710  020650  000405                          BR      6$
7711
7712  020652  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
7713                                                                  ;/COUNT EXPECTED?
7714  020660  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
7715
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7719  020662  104022                          ERROR   22              ;/ERROR CLOCK A 100HZ, MINIMUM
7720                                                                  ;/COUNT NOT REACHED DURING DELAY,
7721                                                                  ;/SECOND COUNT PERIOD.
7722
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7726
7727  020664                          6$:
7728
7729
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 200      *TEST CLOCK B'S 100KHZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 100KHZ DIVIDER WILL DIVIDE 1MHZ
                                        ;*BY 10 TO GIVE US A "100KHZ CLK" L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE,
                                        ;*ENABLE CLOCK B "FEED B TO A" TO ROUTE CLOCK A'S 1MHZ PULSES, AND
                                        ;*GENERATE 10. "1MHZ CLK" H PULSES WHICH GIVES US 10 1MHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100KHZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 1MHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 100KHZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                        ;*
                                        ;;**********************************************************************
020664  000004                          TST200: SCOPE
020666  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

020674  005077  160432                          CLR     @BSR            ;/CLEAR CLOCK B.
020700  005077  160420                          CLR     @ASR            ;/CLEAR CLOCK A.
020704  005077  160424                          CLR     @BBR            ;/CLEAR B'S BUFFER + COUNT REGS.
020710  012777  004040  160414                  MOV     #BIT11!BIT5,@BSR  ;/DISABLE THE 1MHZ OSC.
020716  052777  000004  160406                  BIS     #4,@BSR ;/RATE: 100KHZ
020724  005277  160402                          INC     @BSR            ;/ENABLE CLOCK B.
020730  012700  177766                          MOV     #-10.,R0        ;/SET TO GENERATE 10. 1MHZ PULSES

020734  052777  004000  160362  1$:     BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
020742  005777  160370                  TST     @BCR            ;/HAS THE COUNTER ADVANCED?
020746  001002                          BNE     10$             ;/EXIT LOOP IF SO.
                                                                ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                                ;/      OSC.,THE DIVIDER COULD HAVE
                                                                ;/      HAD ANY COUNT IN IT.
                                                                ;/AFTER THIS LOOP,WE SHOULD BE SUNK.

020750  005200                          INC     R0              ;/DONE 10. 1MHZ PULSES?
020752  001370                          BNE     1$              ;/IF NOT - DO ANOTHER.

020754  012737  000001  001124  10$:     MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

020762  017737  160350  001126          MOV     @BCR,$BDDAT     ;/READ THE COUNTER

020770  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
020776  001402                          BEQ     2$              ;/IF YES - NEXT CHECK

        ;;;$$$
```

```
7788  021002  000417                          BR      4$
7789
7790  021004  012700  177767          2$:     MOV     #-9.,R0 ;/GET THE NUMBER OF "1 MHZ" H PULSES
7791                                                        ;/NEED TO GIVE 9 1MHZ PULSES
7792  021010  052777  004000  160306  3$:     BIS     #BIT11,@ASR     ;/GENERATE 9 1MHZ PULSES
7793  021016  005200                          INC     R0              ;/INORDER TO CHECK TO SEE THAT
7794  021020  001373                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 100KHZ PULSE
7795
7796  021022  017737  160310  001126          MOV     @BCR,$BDDAT     ;/READ THE COUNTER.
7797  021030  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100KHZ PULSE GENERATED?
7798  021036  001401                          BEQ     4$              ;/NO - GOTO NEXT CHECK.
7799
7800

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

7804  021040  104015                  ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
7805                                                        ;/GENERATED A SECOND 100KHZ PULSE
7806                                                        ;/ON ONLY 9 1MHZ PULSES.
7807

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

7811  021042                          4$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 201      *TEST CLOCK B'S 10KHZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 10KHZ DIVIDER WILL DIVIDE 100KHZ
                                        ;*BY 10 TO GIVE US A "10KHZ CLK" L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE,
                                        ;*ENABLE CLOCK B "FEED B TO A" TO ROUTE CLOCK A'S 1MHZ PULSES, AND
                                        ;*GENERATE 100. "1MHZ CLK" H PULSES WHICH GIVES US 10 100KHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 10KHZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 100KHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 10KHZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                        ;*
                                        ;;**********************************************************************
021042  000004                        TST201: SCOPE
021044  012737  000020  001164                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

021052  005077  160254                        CLR     @BSR            ;/CLEAR CLOCK B.
021056  005077  160242                        CLR     @ASR            ;/CLEAR CLOCK A.
021062  005077  160246                        CLR     @BBR            ;/CLEAR B'S BUFFER + COUNT REGS.
021066  012777  004040  160236                MOV     #BIT11!BIT5,@BSR  ;/DISABLE THE 1MHZ OSC.
021074  052777  000006  160230                BIS     #6,@BSR ;/RATE: 10KHZ
021102  005277  160224                        INC     @BSR            ;/ENABLE CLOCK B.
021106  012700  177634                        MOV     #-100.,R0       ;/SET TO GENERATE 100. 1MHZ PULSES

021112  052777  004000  160204  1$:   BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
021120  005777  160212                TST     @BCR            ;/HAS THE COUNTER ADVANCED?
021124  001002                        BNE     10$             ;/EXIT LOOP IF SO.
                                                              ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                              ;/      OSC.,THE DIVIDER COULD HAVE
                                                              ;/      HAD ANY COUNT IN IT.
                                                              ;/AFTER THIS LOOP,WE SHOULD BE SUNK.

021126  005200                        INC     R0              ;/DONE 100. 1MHZ PULSES?
021130  001370                        BNE     1$              ;/IF NOT - DO ANOTHER.

021132  012737  000001  001124  10$:  MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

021140  017737  160172  001126        MOV     @BCR,$BDDAT     ;/READ THE COUNTER

021146  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
021154  001402                        BEQ     2$              ;/IF YES - NEXT CHECK

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS

021156  104015                        ERROR   15              ;/ERROR - CLOCK B - 10KHZ - PULSE
                                                              ;/NOT GENERATED WHEN 10 100KHZ PULSE
                                                              ;/WERE GENERATED.

        ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSS
```

```
7870  021160  000417                          BR      4$
7871
7872  021162  012700  177635          2$:     MOV     #-99.,R0        ;/GET THE NUMBER OF "1 MHZ" H PULSES
7873                                                                  ;/NEED TO GIVE 9 100KHZ PULSES
7874  021166  052777  004000  160130  3$:     BIS     #BIT11,@ASR     ;/GENERATE 9 100KHZ PULSES
7875  021174  005200                          INC     R0              ;/INORDER TO CHECK TO SEE THAT
7876  021176  001373                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 10KHZ PULSE
7877
7878  021200  017737  160132  001126          MOV     @BCR,$BDDAT     ;/READ THE COUNTER.
7879  021206  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 10KHZ PULSE GENERATED?
7880  021214  001401                          BEQ     4$              ;/NO - GOTO NEXT CHECK.
7881
7882

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

7886  021216  104015                          ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
7887                                                                  ;/GENERATED A SECOND 10KHZ PULSE
7888                                                                  ;/ON ONLY 9 100KHZ PULSES.
7889

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

7893  021220                          4$:
```

```
                                        ;/*
                                ;;*****************************************************************
                                ;*TEST 202       *TEST CLOCK B'S 1KHZ DIVIDER
                                ;*
                                ;*IN THIS TEST WE'LL SEE IF THE 1KHZ DIVIDER WILL DIVIDE 10KHZ
                                ;*BY 10 TO GIVE US A "1KHZ CLK" L PULSE.
                                ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE,
                                ;*ENABLE CLOCK B "FEED B TO A" TO ROUTE CLOCK A'S 1MHZ PULSES, AND
                                ;*GENERATE 1000. "1MHZ CLK" H PULSES WHICH GIVES US 10 10KHZ
                                ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 1KHZ
                                ;*PULSES.
                                ;*THEN WE'LL GENERATE 9 MORE 10KHZ PULSES AND MAKE
                                ;*SURE THAT WE DON'T GET ANOTHER 1KHZ PULSE.
                                ;*
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                ;*
                                ;*****************************************************************
021220  000004                  TST202: SCOPE
021222  012737  000020  001164          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
021230  005077  160076          CLR     @BSR            ;/CLEAR CLOCK B.
021234  005077  160064          CLR     @ASR            ;/CLEAR CLOCK A.
021240  005077  160070          CLR     @BBR            ;/CLEAR B'S BUFFER + COUNT REGS.
021244  012777  004040  160060  MOV     #BIT11!BIT5,@BSR  ;/DISABLE THE 1MHZ OSC.
021252  052777  000010  160052  BIS     #10,@BSR        ;/RATE: 1KHZ
021260  005277  160046          INC     @BSR            ;/ENABLE CLOCK B.
021264  012700  176030          MOV     #-1000.,R0      ;/SET TO GENERATE 1000. 1MHZ PULSES
021270  052777  004000  160026  1$:     BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
021276  005777  160034          TST     @BCR            ;/HAS THE COUNTER ADVANCED?
021302  001002                  BNE     10$             ;/EXIT LOOP IF SO.
                                                        ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                        ;/      OSC., THE DIVIDER COULD HAVE
                                                        ;/      HAD ANY COUNT IN IT.
                                                        ;/AFTER THIS LOOP,WE SHOULD BE SUNK.
021304  005200                  INC     R0              ;/DONE 1000. 1MHZ PULSES?
021306  001370                  BNE     1$              ;/IF NOT - DO ANOTHER.
021310  012737  000001  001124  10$:    MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
021316  017737  160014  001126          MOV     @BCR,$BDDAT     ;/READ THE COUNTER
021324  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
021332  001402                  BEQ     2$              ;/IF YES - NEXT CHECK

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

021334  104015                  ERROR   15              ;/ERROR - CLOCK B - 1KHZ - PULSE
                                                        ;/NOT GENERATED WHEN 10 10KHZ PULSE
                                                        ;/WERE GENERATED.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS
```

```
7952  021336  000417                          BR      4$
7953
7954  021340  012700  176031          2$:     MOV     #-999.,R0       ;/GET THE NUMBER OF "1 MHZ" H PULSES
7955                                                                    ;/NEED TO GIVE 9 10KHZ PULSES
7956  021344  052777  004000  157752  3$:     BIS     #BIT11,@ASR     ;/GENERATE 9 10KHZ PULSES
7957  021352  005200                          INC     R0              ;/INORDER TO CHECK TO SEE THAT
7958  021354  001373                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 1KHZ PULSE
7959
7960  021356  017737  157754  001126          MOV     @BCR,$BDDAT     ;/READ THE COUNTER.
7961  021364  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 1KHZ PULSE GENERATED?
7962  021372  001401                          BEQ     4$              ;/NO - GOTO NEXT CHECK.
7963
7964

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7968  021374  104015                          ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
7969                                                                    ;/GENERATED A SECOND 1KHZ PULSE
7970                                                                    ;/ON ONLY 9 10KHZ PULSES.
7971

                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7975  021376                        4$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 203      *TEST CLOCK B'S 100HZ DIVIDER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF THE 100HZ DIVIDER WILL DIVIDE 1KHZ
                                        ;*BY 10 TO GIVE US A "100HZ CLK" L PULSE.
                                        ;*TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE,
                                        ;*ENABLE CLOCK B "FEED B TO A" TO ROUTE CLOCK A'S 1MHZ PULSES, AND
                                        ;*GENERATE 10000. "1MHZ CLK" H PULSES WHICH GIVES US 10 1KHZ
                                        ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100HZ
                                        ;*PULSES.
                                        ;*THEN WE'LL GENERATE 9 MORE 1KHZ PULSES AND MAKE
                                        ;*SURE THAT WE DON'T GET ANOTHER 100HZ PULSE.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                        ;*
                                        ;**********************************************************************
021376  000004                        TST203: SCOPE
021400  012737  000020  001164                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

021406  005077  157720                        CLR     @BSR            ;/CLEAR CLOCK B.
021412  005077  157706                        CLR     @ASR            ;/CLEAR CLOCK A.
021416  005077  157712                        CLR     @BBR            ;/CLEAR B'S BUFFER + COUNT REGS.
021422  012777  004040  157702                MOV     #BIT11!BIT5,@BSR  ;/DISABLE THE 1MHZ OSC.
021430  052777  000012  157674                BIS     #12,@BSR        ;/RATE: 100HZ
021436  005277  157670                        INC     @BSR            ;/ENABLE CLOCK B.
021442  012700  154360                        MOV     #-10000.,R0     ;/SET TO GENERATE 10000. 1MHZ PULSES

021446  052777  004000  157650  1$:   BIS     #BIT11,@ASR     ;/GENERATE 1 1MHZ PULSE
021454  005777  157656                TST     @BCR            ;/HAS THE COUNTER ADVANCED?
021460  001002                        BNE     10$             ;/EXIT LOOP IF SO.
                                                              ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
                                                              ;/      OSC., THE DIVIDER COULD HAVE
                                                              ;/      HAD ANY COUNT IN IT.
                                                              ;/AFTER THIS LOOP,WE SHOULD BE SUNK.

021462  005200                        INC     R0              ;/DONE 10000. 1MHZ PULSES?
021464  001370                        BNE     1$              ;/IF NOT - DO ANOTHER.

021466  012737  000001  001124  10$:  MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.

021474  017737  157636  001126        MOV     @BCR,$BDDAT     ;/READ THE COUNTER

021502  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
021510  001402                        BEQ     2$              ;/IF YES - NEXT CHECK

        ;;;$$$
```

```
8034  021514  000417                          BR      4$
8035
8036  021516  012700  154361          2$:     MOV     #-9999.,R0      ;/GET THE NUMBER OF "1 MHZ" H PULSES
8037                                                                  ;/NEED TO GIVE 9 1KHZ PULSES
8038  021522  052777  004000  157574  3$:     BIS     #BIT11,@ASR     ;/GENERATE 9 1KHZ PULSES
8039  021530  005200                          INC     R0              ;/INORDER TO CHECK TO SEE THAT
8040  021532  001373                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 100HZ PULSE
8041
8042  021534  017737  157576  001126          MOV     @BCR,$BDDAT     ;/READ THE COUNTER.
8043  021542  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100HZ PULSE GENERATED?
8044  021550  001401                          BEQ     4$              ;/NO - GOTO NEXT CHECK.
8045
8046

            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

8050  021552  104015                          ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
8051                                                                  ;/GENERATED A SECOND 100HZ PULSE
8052                                                                  ;/ON ONLY 9 1KHZ PULSES.
8053

            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

8057  021554                          4$:
8058
8059
8060
8061
8062
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 204      *TEST CLOCK B'S REPEATIBILITY AT 1MHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 1MHZ REPEATABILITY OF
                                        ;*CLOCK B (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +- 2.)
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;;*****************************************************************
021554  000004                          TST204: SCOPE
021556  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

021564  005077  157534                          CLR     @ASR            ;/CLEAR CLOCK A.
021570  005077  157536                          CLR     @BSR            ;/CLEAR CLOCK B.
021574  005077  157534                          CLR     @BBR            ;/CLEAR CLOCK B'S BUFFER REG.
021600  012777  000002  157524                  MOV     #2,@BSR ;/SET RATE: 1MHZ.
021606  012700  000036                          MOV     #30.,R0 ;/SET THE DELAY.

021612  005277  157514                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

021616  005300                          1$:     DEC     R0              ;/DELAYL
021620  001376                                  BNE     1$

021622  042777  000016  157502                  BIC     #16,@BSR        ;/STOP THE CLOCK.
021630  017737  157502  001124                  MOV     @BCR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

021636  005077  157472                          CLR     @BBR            ;/RELOAD THE BUF. REG.
021642  012777  000002  157462                  MOV     #2,@BSR ;/SET RATE: 1MHZ.
021650  012700  000036                          MOV     #30.,R0 ;/SET THE DELAY.

021654  005277  157452                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

021660  005300                          2$:     DEC     R0              ;/DELAY (SAME AS AT 1$)
021662  001376                                  BNE     2$

021664  042777  000016  157440                  BIC     #16,@BSR        ;/STOP THE CLOCK!
021672  017737  157440  001126                  MOV     @BCR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

021700  013700  001124                          MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
021704  163700  001126                          SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                        ;/IN ORDER TO FIND OUT WHAT THE
                                                                        ;/VARIENCE WAS.
021710  100001                                  BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
021712  005400                                  NEG     R0              ;/DO IF SUB WAS A NEG RESULT
                                                                        ;/MAKE IT POSITIVE.
021714                                  3$:
021714  020027  000002                          CMP     R0,#2   ;/DID THE TWO COUNT UPS VARY
                                                                        ;/MORE THAN 2?
021720  003402                                  BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8122
 8123  021722  104023                          ERROR   23              ;/ERROR - CLOCK B - 1MHZ REPEATABILITY
 8124                                                                  ;/VARIANCE GREATER THAN 2.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8128  021724  000416                          BR      6$
 8129
 8130  021726  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.
 8131
 8132  021734  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
 8133                                                                  ;/COUNT EXPECTED?
 8134
 8135  021742  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
 8136
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8140  021744  104024                          ERROR   24              ;/ERROR CLOCK B 1MHZ, MINIMUM
 8141                                                                  ;/COUNT NOT REACHED DURING DELAY
 8142                                                                  ;/1ST COUNT PERIOD.
 8143
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8147  021746  000405                          BR      6$
 8148
 8149  021750  023737  001162  001126  5$:     CMP     $TMPO,$BODAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
 8150                                                                  ;/COUNT EXPECTED?
 8151  021756  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
 8152
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8156  021760  104025                          ERROR   25              ;/ERROR CLOCK B 1MHZ, MINIMUM
 8157                                                                  ;/COUNT NOT REACHED DURING DELAY
 8158                                                                  ;/SECOND COUNT PERIOD.
 8159
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8163  021762  005077  157344          6$:     CLR     @BSR            ;/LEAR THE CLOCK.
 8164
 8165
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 205      *TEST CLOCK B'S REPEATIBILITY AT 100HKZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 100HKZ REPEATABILITY OF
                                        ;*CLOCK B (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +- 2.)
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;;*****************************************************************
021766  000004                          TST205: SCOPE
021770  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

021776  005077  157322                          CLR     @ASR            ;/CLEAR CLOCK A.
022002  005077  157324                          CLR     @BSR            ;/CLEAR CLOCK B.
022006  005077  157322                          CLR     @BBR            ;/CLEAR CLOCK B'S BUFFER REG.
022012  012777  000004  157312                  MOV     #4,@BSR ;/SET RATE: 100HKZ.
022020  012700  000454                          MOV     #300.,R0        ;/SET THE DELAY.

022024  005277  157302                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022030  005300                          1$:     DEC     R0              ;/DELAYL
022032  001376                                  BNE     1$

022034  042777  000016  157270                  BIC     #16,@BSR        ;/STOP THE CLOCK.
022042  017737  157270  001124                  MOV     @BCR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

022050  005077  157260                          CLR     @BBR            ;/RELOAD THE BUF. REG.
022054  012777  000004  157250                  MOV     #4,@BSR ;/SET RATE: 100HKZ.
022062  012700  000454                          MOV     #300.,R0        ;/SET THE DELAY.

022066  005277  157240                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022072  005300                          2$:     DEC     R0              ;/DELAY (SAME AS AT 1$)
022074  001376                                  BNE     2$

022076  042777  000016  157226                  BIC     #16,@BSR        ;/STOP THE CLOCK!
022104  017737  157226  001126                  MOV     @BCR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

022112  013700  001124                          MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
022116  163700  001126                          SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                        ;/IN ORDER TO FIND OUT WHAT THE
                                                                        ;/VARIENCE WAS.
022122  100001                                  BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
022124  005400                                  NEG     R0              ;/DO IF SUB WAS A NEG RESULT
                                                                        ;/MAKE IT POSITIVE.
022126                                  3$:
022126  020027  000002                          CMP     R0,#2   ;/DID THE TWO COUNT UPS VARY
                                                                        ;/MORE THAN 2?
022132  003402                                  BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8225
 8226  022134  104023                          ERROR   23              ;/ERROR - CLOCK B - 100HKZ REPEATABILITY
 8227                                                                  ;/VARIANCE GREATER THAN 2.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8231  022136  000416                          BR      6$
 8232
 8233  022140  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.
 8234
 8235  022146  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
 8236                                                                  ;/COUNT EXPECTED?
 8237
 8238  022154  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
 8239
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8243  022156  104024                          ERROR   24              ;/ERROR CLOCK B 100HKZ, MINIMUM
 8244                                                                  ;/COUNT NOT REACHED DURING DELAY
 8245                                                                  ;/1ST COUNT PERIOD.
 8246
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8250  022160  000405                          BR      6$
 8251
 8252  022162  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
 8253                                                                  ;/COUNT EXPECTED?
 8254  022170  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
 8255
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8259  022172  104025                          ERROR   25              ;/ERROR CLOCK B 100HKZ, MINIMUM
 8260                                                                  ;/COUNT NOT REACHED DURING DELAY
 8261                                                                  ;/SECOND COUNT PERIOD.
 8262
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8266  022174  005077  157132          6$:     CLR     @BSR            ;/LEAR THE CLOCK.
 8267
 8268
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 206      *TEST CLOCK B'S REPEATIBILITY AT 10KHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 10KHZ REPEATABILITY OF
                                        ;*CLOCK B (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +- 2.)
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;;*****************************************************************
022200  000004                          TST206: SCOPE
022202  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

022210  005077  157110                          CLR     @ASR            ;/CLEAR CLOCK A.
022214  005077  157112                          CLR     @BSR            ;/CLEAR CLOCK B.
022220  005077  157110                          CLR     @BBR            ;/CLEAR CLOCK B'S BUFFER REG.
022224  012777  000006  157100                  MOV     #6,@BSR ;/SET RATE: 10KHZ.
022232  012700  005670                          MOV     #3000.,R0       ;/SET THE DELAY.

022236  005277  157070                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022242  005300                          1$:     DEC     R0              ;/DELAYL
022244  001376                                  BNE     1$

022246  042777  000016  157056                  BIC     #16,@BSR        ;/STOP THE CLOCK.
022254  017737  157056  001124                  MOV     @BCR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

022262  005077  157046                          CLR     @BBR            ;/RELOAD THE BUF. REG.
022266  012777  000006  157036                  MOV     #6,@BSR ;/SET RATE: 10KHZ.
022274  012700  005670                          MOV     #3000.,R0       ;/SET THE DELAY.

022300  005277  157026                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022304  005300                          2$:     DEC     R0              ;/DELAY (SAME AS AT 1$)
022306  001376                                  BNE     2$

022310  042777  000016  157014                  BIC     #16,@BSR        ;/STOP THE CLOCK!
022316  017737  157014  001126                  MOV     @BCR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

022324  013700  001124                          MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
022330  163700  001126                          SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                        ;/IN ORDER TO FIND OUT WHAT THE
                                                                        ;/VARIENCE WAS.
022334  100001                                  BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
022336  005400                                  NEG     R0              ;/DO IF SUB WAS A NEG RESULT
                                                                        ;/MAKE IT POSITIVE.
022340                                  3$:
022340  020027  000002                          CMP     R0,#2   ;/DID THE TWO COUNT UPS VARY
                                                                        ;/MORE THAN 2?
022344  003402                                  BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8328
 8329  022246  104023                          ERROR   23              ;/ERROR - CLOCK B - 10KHZ REPEATABILITY
 8330                                                                  ;/VARIANCE GREATER THAN 2.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8334  022350  000416                          BR      6$
 8335
 8336  022352  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.
 8337
 8338  022360  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
 8339                                                                  ;/COUNT EXPECTED?
 8340
 8341  022366  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
 8342
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8346  022370  104024                          ERROR   24              ;/ERROR CLOCK B 10KHZ, MINIMUM
 8347                                                                  ;/COUNT NOT REACHED DURING DELAY
 8348                                                                  ;/1ST COUNT PERIOD.
 8349
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8353  022372  000405                          BR      6$
 8354
 8355  022374  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
 8356                                                                  ;/COUNT EXPECTED?
 8357  022402  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
 8358
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8362  022404  104025                          ERROR   25              ;/ERROR CLOCK B 10KHZ, MINIMUM
 8363                                                                  ;/COUNT NOT REACHED DURING DELAY
 8364                                                                  ;/SECOND COUNT PERIOD.
 8365
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8369  022406  005077  156720          6$:     CLR     @BSR            ;/LEAR THE CLOCK.
 8370
 8371
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 207      *TEST CLOCK B'S REPEATIBILITY AT 1KHZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 1KHZ REPEATABILITY OF
                                        ;*CLOCK B (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +- 2.)
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;;**********************************************************************
022412  000004                          ST207: SCOPE
022414  012737  000020  001164                 MOV     #20,$TIMES      ;;DO 20 ITERATIONS

022422  005077  156676                       CLR     @ASR            ;/CLEAR CLOCK A.
022426  005077  156700                       CLR     @BSR            ;/CLEAR CLOCK B.
022432  005077  156676                       CLR     @BBR            ;/CLEAR CLOCK B'S BUFFER REG.
022436  012777  000010  156666               MOV     #10,@BSR        ;/SET RATE: 1KHZ.
022444  012700  072460                       MOV     #30000.,R0      ;/SET THE DELAY.

022450  005277  156656                       INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022454  005300                        1$:    DEC     R0              ;/DELAYL
022456  001376                               BNE     1$

022460  042777  000016  156644               BIC     #16,@BSR        ;/STOP THE CLOCK.
022466  017737  156644  001124               MOV     @BCR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

022474  005077  156634                       CLR     @BBR            ;/RELOAD THE BUF. REG.
022500  012777  000010  156624               MOV     #10,@BSR        ;/SET RATE: 1KHZ.
022506  012700  072460                       MOV     #30000.,R0      ;/SET THE DELAY.

022512  005277  156614                       INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022516  005300                        2$:    DEC     R0              ;/DELAY (SAME AS AT 1$)
022520  001376                               BNE     2$

022522  042777  000016  156602               BIC     #16,@BSR        ;/STOP THE CLOCK!
022530  017737  156602  001126               MOV     @BCR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

022536  013700  001124                       MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
022542  163700  001126                       SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                     ;/IN ORDER TO FIND OUT WHAT THE
                                                                     ;/VARIENCE WAS.
022546  100001                               BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
022550  005400                               NEG     R0              ;/DO IF SUB WAS A NEG RESULT
                                                                     ;/MAKE IT POSITIVE.
022552                                  3$:
022552  020027  000002                       CMP     R0,#2   ;/DID THE TWO COUNT UPS VARY
                                                                     ;/MORE THAN 2?
022556  003402                               BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 8431
 8432  022560  104023                          ERROR   23              ;/ERROR - CLOCK B - 1KHZ REPEATABILITY
 8433                                                                  ;/VARIANCE GREATER THAN 2.

                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 8437  022562  000416                          BR      6$
 8438
 8439  022564  012737  000001  001162  4$:     MOV     #1,$TMPO        ;/SET MINIMUM COUNT EXPECTED.
 8440
 8441  022572  023737  001162  001124          CMP     $TMPO,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
 8442                                                                  ;/COUNT EXPECTED?
 8443
 8444  022600  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
 8445
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 8449  022602  104024                          ERROR   24              ;/ERROR CLOCK B 1KHZ, MINIMUM
 8450                                                                  ;/COUNT NOT REACHED DURING DELAY
 8451                                                                  ;/1ST COUNT PERIOD.
 8452
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 8456  022604  000405                          BR      6$
 8457
 8458  022606  023737  001162  001126  5$:     CMP     $TMPO,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
 8459                                                                  ;/COUNT EXPECTED?
 8460  022614  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
 8461
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 8465  022616  104025                          ERROR   25              ;/ERROR CLOCK B 1KHZ, MINIMUM
 8466                                                                  ;/COUNT NOT REACHED DURING DELAY
 8467                                                                  ;/SECOND COUNT PERIOD.
 8468
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

 8472  022620  005077  156506          6$:     CLR     @BSR            ;/LEAR THE CLOCK.
 8473
 8474
```

```
                                                ;/#

                                        ;;*****************************************************************
                                        ;*TEST 210      *TEST CLOCK B'S REPEATIBILITY AT 100HZ RATE
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK 100HZ REPEATABILITY OF
                                        ;*CLOCK B (THE ABILITY OF THE CLOCK TO COUNT TO THE SAME
                                        ;*VALUE DURING THE SAME TIME SPACE TWICE +- 2.)
                                        ;*WE'LL ALSO CHECK TO MAKE SURE THAT IT MAKES 1 COUNT(S)
                                        ;*DURING THE TIME PERIOD.
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                        ;*
                                        ;;*****************************************************************
022624  000004                          TST210: SCOPE
022626  012737  000020  001164                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

022634  005077  156464                          CLR     @ASR            ;/CLEAR CLOCK A.
022640  005077  156466                          CLR     @BSR            ;/CLEAR CLOCK B.
022644  005077  156464                          CLR     @BBR            ;/CLEAR CLOCK B'S BUFFER REG.
022650  012777  000012  156454                  MOV     #12,@BSR        ;/SET RATE: 100HZ.
022656  012700  177777                          MOV     #-1,R0  ;/SET THE DELAY.

022662  005277  156444                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022666  005300                          1$:     DEC     R0              ;/DELAYL
022670  001376                                  BNE     1$

022672  042777  000016  156432                  BIC     #16,@BSR        ;/STOP THE CLOCK.
022700  017737  156432  001124                  MOV     @BCR,$GDDAT     ;/READ THE COUNTER, STORE IN "$GDDAT".

022706  005077  156422                          CLR     @BBR            ;/RELOAD THE BUF. REG.
022712  012777  000012  156412                  MOV     #12,@BSR        ;/SET RATE: 100HZ.
022720  012700  177777                          MOV     #-1,R0  ;/SET THE DELAY.

022724  005277  156402                          INC     @BSR            ;/ENABLE THE COUNTER TO COUNT.

022730  005300                          2$:     DEC     R0              ;/DELAY (SAME AS AT 1$)
022732  001376                                  BNE     2$

022734  042777  000016  156370                  BIC     #16,@BSR        ;/STOP THE CLOCK!
022742  017737  156370  001126                  MOV     @BCR,$BDDAT     ;/READ THE COUNTER, STORE IN "$BDDAT".

022750  013700  001124                          MOV     $GDDAT,R0       ;/GET FIRST COUNT VALUE.
022754  163700  001126                          SUB     $BDDAT,R0       ;/SUBTRACT THE SECOND COUNT VALUE
                                                                        ;/IN ORDER TO FIND OUT WHAT THE
                                                                        ;/VARIENCE WAS.
022760  100001                                  BPL     3$              ;/NOW WE WANT A POSITIVE VALUE
022762  005400                                  NEG     R0              ;/DO IF SUB WAS A NEG RESULT
                                                                        ;/MAKE IT POSITIVE.
022764                                  3$:
022764  020027  000002                          CMP     R0,#2   ;/DID THE TWO COUNT UPS VARY
                                                                        ;/MORE THAN 2?
022770  003402                                  BLE     4$              ;/NO - NEXT CHECK
```

```
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8534
8535  022772  104023                          ERROR   23              ;/ERROR - CLOCK B - 100HZ REPEATABILITY
8536                                                                  ;/VARIANCE GREATER THAN 2.
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8540  022774  000416                          BR      6$
8541
8542  022776  012737  000001  001162  4$:     MOV     #1,$TMP0        ;/SET MINIMUM COUNT EXPECTED.
8543
8544  023004  023737  001162  001124          CMP     $TMP0,$GDDAT    ;/DID 1ST COUNT = OR EXCEED MINIMUM
8545                                                                  ;/COUNT EXPECTED?
8546
8547  023012  003402                          BLE     5$              ;/BR IF YES - NEXT CHECK
8548
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8552  023014  104024                          ERROR   24              ;/ERROR CLOCK B 100HZ, MINIMUM
8553                                                                  ;/COUNT NOT REACHED DURING DELAY
8554                                                                  ;/1ST COUNT PERIOD.
8555
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8559  023016  000405                          BR      6$
8560
8561  023020  023737  001162  001126  5$:     CMP     $TMP0,$BDDAT    ;/DID 2ND COUNT = OR EXCEED MINIMUM
8562                                                                  ;/COUNT EXPECTED?
8563  023026  003401                          BLE     6$              ;/BR IF YES - NEXT TEST.
8564
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8568  023030  104025                          ERROR   25              ;/ERROR CLOCK B 100HZ, MINIMUM
8569                                                                  ;/COUNT NOT REACHED DURING DELAY
8570                                                                  ;/SECOND COUNT PERIOD.
8571
                ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8575  023032  005077  156274          6$:     CLR     @#BSR           ;/LEAR THE CLOCK.
8576
8577
8578                                          ;
8579                                          ;NOW WE'LL DETERMINE IF WE ARE ALLOWED TO AUTO-SIZE.
8580                                          ;IF SO, WE'LL FIND OUT IF THERE ARE OTHER CLOCKS OUT THERE
8581                                          ;TO TEST.
8582                                          ;
8583
8584  023036  000004                  TE:     SCOPE
8585  023040  105737  001221                  TSTB    $ENVM           ;SEE IF APT WILL LET US AUTO-SIZE.
8586  023044  100521                          BMI     2$              ;NO-EXIT.
```

```
8587
8588  023046  062737  000040  001324          ADD     #40,ASR          ;YES,ADD TO BASE ADDR.
8589  023054  013746  000004                  MOV     ERRVEC,-(6)      ;SAVE CONTENTS OF LOC 4.
8590  023060  012737  023220  000004          MOV     #1$,ERRVEC       ;SET UP IN CASE NO MORE CLOCKS.
8591
8592  023066  005777  156232                  TST     @ASR             ;TIME OUT HERE IF NO MORE CLOCKS.
8593
8594                                                                   ;IF HERE,ANOTHER CLOCK FOUND.
8595  023072  104400  023100                  TYPE    ,65$             ;;TYPE ASCIZ STRING
8596  023076  000405                          BR      64$              ;;GET OVER THE ASCIZ
8597                                  ;;65$:  .ASCIZ  <15><12>"UNIT #"
8598  023112                          64$:
8599  023112  013746  001210                  MOV     $DEVCT,-(SP)     ;;SAVE $DEVCT FOR TYPEOUT
8600  023116  104401                          TYPOC                    ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8601  023120  104400  023126                  TYPE    ,67$             ;;TYPE ASCIZ STRING
8602  023124  000406                          BR      66$              ;;GET OVER THE ASCIZ
8603                                  ;;67$:  .ASCIZ  " COMPLETED "
8604  023142                          66$:
8605  023142  005237  001210                  INC     $DEVCT
8606  023146  104400  023154                  TYPE    ,69$             ;;TYPE ASCIZ STRING
8607  023152  000410                          BR      68$              ;;GET OVER THE ASCIZ
8608                                  ;;69$:  .ASCIZ  " TESTING UNIT #"
8609  023174                          68$:
8610  023174  013746  001210                  MOV     $DEVCT,-(SP)     ;;SAVE $DEVCT FOR TYPEOUT
8611  023200  104401                          TYPOC                    ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8612  023202  012637  000004                  MOV     (6)+,ERRVEC      ;RESTORE LOC 4.
8613  023206  062737  000040  001340          ADD     #40,AVECT        ;UPDATE VECTOR ADDR.
8614  023214  000137  002174                  JMP     LOOP             ;TEST NEW UNIT.
8615
8616  023220                          1$:
8617  023220  062706  000004                  ADD     #4,R6            ;ADD #4 TO THE STACK POINTER
8618  023224  012637  000004                  MOV     (6)+,ERRVEC      ;RESTORE LOC 4
8619  023230  022737  000001  001210          CMP     #1,$DEVCT        ;TESTED ONLY ONE UNIT?
8620  023236  001424                          BEQ     2$               ;YES-NO NEED FOR TYPEOUT.
8621
8622  023240  104400  023246                  TYPE    ,71$             ;;TYPE ASCIZ STRING
8623  023244  000405                          BR      70$              ;;GET OVER THE ASCIZ
8624                                  ;;71$:  .ASCIZ  <15><12>"UNIT #"
8625  023260                          70$:
8626  023260  013746  001210                  MOV     $DEVCT,-(SP)     ;;SAVE $DEVCT FOR TYPEOUT
8627  023264  104401                          TYPOC                    ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8628  023266  104400  023274                  TYPE    ,73$             ;;TYPE ASCIZ STRING
8629  023272  000406                          BR      72$              ;;GET OVER THE ASCIZ
8630                                  ;;73$:  .ASCIZ  " COMPLETED "
8631  023310                          72$:
8632
8633  023310  013737  001254  001324  2$:     MOV     $BASE,ASR
8634  023316  013737  001250  001340          MOV     $VECT1,AVECT
8635  023324  013737  001252  001350          MOV     $PRIOR,APRITY
8636  023332  012737  000001  001210          MOV     #1,$DEVCT
8637
8638                                          .SBTTL
8639
8640
8641                                  .SBTTL  END OF PASS ROUTINE
8642
```

```
 8643                                   ;;**********************************************************************
 8644                                   ;*INCREMENT THE PASS NUMBER ($PASS)
 8645                                   ;*TYPE "END PASS"
 8646                                   ;*IF THERES A MONITOR GO TO IT
 8647                                   ;*IF THERE ISN'T JUMP TO LOOP
 8648                                   ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
 8649                                   ;*SENDMG CAN BE CHANGED TO 7.
 8650
 8651  023340                           $EOP:
 8652  023340  000240                          NOP
 8653  023342  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 8654  023346  005037  001164                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 8655  023352  005237  001206                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
 8656  023356  042737  100000  001206        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 8657  023364  005327                          DEC     (PC)+           ;;LOOP?
 8658  023366  000001                $EOPCT: .WORD   1
 8659  023370  003015                          BGT     $DOAGN          ;;YES
 8660  023372  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 8661  023374  000001                $ENDCT: .WORD   1
 8662  023376  023366                          $EOPCT
 8663  023400  104400  023433                  TYPE    ,$ENDMG         ;;TYPE "END PASS"
 8664  023404  013700  000042        $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 8665  023410  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 8666  023412  000005                          RESET                   ;;CLEAR THE WORLD
 8667  023414  004710                $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 8668  023416  000240                          NOP                     ;;SAVE ROOM
 8669  023420  000240                          NOP                     ;;FOR
 8670  023422  000240                          NOP                     ;;ACT11
 8671  023424                        $DOAGN:
 8672  023424  000137                          JMP     @(PC)+          ;;RETURN
 8673  023426  002174                $RTNAD: .WORD   LOOP
 8674  023430     377       377     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 8675  023433     015    042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS/
 8676  023440  050040  051501  000123
 8677
 8678                                   .SBTTL  *
```

```
                                        .SBTTL  * SPECIAL I/O SIGNAL TESTS
                                        .SBTTL  *



                                .SBTTL  *       "STP2 OUT" TO "SCHMITT TRIG 1 IN" TESTS
                                        ;*
                                        ;* THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND
                                        ;* PROVIDING SCOPE LOOP CAPABILITIES FOR "STP2 OUT" L
                                        ;* AND "SCHMITT TRIG 1" IN.
                                        ;*
                                        ;* WHEN YOU LOAD AND START AT LOCATION 210, PROGRAM
                                        ;* CONTROL IS TRANSFERRED HERE. "STP2 OUT" L PULSES ARE
                                        ;* GENERATED BY "LO STAT A HI" H + "BD10" H (MAIN. STP2).
                                        ;* PIN V ("STP2 OUT") IS WIRED TO PIN LL (SCHMITT TRIG1) FOR
                                        ;* THIS TEST. "STP2 OUT" PULSES ARE RECEIVED AS "SCHMITT TRIG 1"
                                        ;* PULSES WHICH SET CLOCK A'S STATUS REGISTER BIT 15.
                                        ;* IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIC.
                                        ;* AND ERROR SWITCH REGISTER OPTIONS ARE USED.
                                        ;* AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE
                                        ;* TEST. SW13=1 WILL INHIBIT THIS FEATURE.
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                ;*
                                        ;*
                                        ;* YOU MUST WIRE PINS V AND LL OF J1 TOGETHER.
                                        ;*
                                        ;* LOGIC TEST (L + S 200) SHOULD BE RUN FIRST.
                                        ;*

023446                          LS210:

023446  005037  001104          1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
023452  012737  177704  001206          MOV     #-60.,$PASS     ;/SET PASS COUNT.
                                                                ;/NOTE: PASS COUNT USED ONLY
                                                                ;/      TO DETECT 60 PASSES SO
                                                                ;/      IT CAN GENERATE A CRLF.
                                                                ;/      AFTER CRLF IT WILL BE ZEROED.
023460  005077  155640          2$:     CLR     @ASR            ;/CLEAR CLOCK A.
023464  005077  155642                  CLR     @BSR            ;/CLEAR CLOCK B.

023470  052777  002000  155626          BIS     #BIT10,@ASR     ;GENERATE AN "STP2 OUT" PULSE
                                                                ;AT THIS POINT YOU SHOULD
                                                                ;SEE AN OUTPUT AT PIN V.
                                                                ;PIN V SHOULD BE WIRED TO
                                                                ;PIN LL FOR "SCHMITT TRIG 1" IN.

023476  032777  100000  155620          BIT     #BIT15,@ASR     ;IS "ST1 FLAG" BIT 15 IN CLOCK
                                                                ;A'S CSR SET?
023504  001001                          BNE     3$              ;BR IF YES TO 3$:

                ;;;SSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSS
```

```
8735
8736  023506  104000                          ERROR                   ;ERROR "SCHMITT TRIG 1" IN NOT
8737                                                                  ;RECEIVED. HAVE YOU WIRED IT RIGHT?
8738
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

8742  023510                          3$:
8743
8744  023510  032777  020000  155420            BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
8745  023516  001360                            BNE     2$              ;/YES - IGNORE ANY UPDATES.
8746
8747  023520  005237  001104                    INC     $ICNT           ;/UPDATE COUNT.
8748  023524  001355                            BNE     2$              ;/IF NOT DONE 65,324 TIMES,
8749                                                                  ;/DO IT AGAIN.
8750
8751  023526  104400  023534                    TYPE    ,65$            ;;TYPE ASCIZ STRING
8752  023532  000401                            BR      64$             ;;GET OVER THE ASCIZ
8753                                  ;;65$:    .ASCIZ  #*#
8754  023536                          64$:
8755
8756  023536  005237  001206                    INC     $PASS           ;/DONE 60 PASSES?
8757  023542  100746                            BMI     2$              ;/NO - NO NEED FOR CR,LF.
8758  023544  104400  023552                    TYPE    ,67$            ;;TYPE ASCIZ STRING
8759  023550  000402                            BR      66$             ;;GET OVER THE ASCIZ
8760                                  ;;67$:    .ASCIZ  <15><12>#
8761  023556                          66$:
8762  023556  000733                            BR      1$
8763
8764
8765                                .SBTTL  ;*      "STP1 OUT" TO "SCHMITT TRIG 2" H TESTS
8766                                        ;*
8767                                        ;* THIS IS A SPECIAL TEST SECTION DEVOTED FOR TESTING AND
8768                                        ;* PROVIDING SCOPE LOOP CAPABILITIES FOR "STP1 OUT" AND
8769                                        ;* "SCHMITT TRIG2" IN.
8770                                        ;*
8771                                        ;* WHEN YOU LOAD AND START AT LOCATION 214, PROGRAM
8772                                        ;* CONTROL IS TRANSFERRED HERE. "STP1 OUT" L PULSES ARE
8773                                        ;* GENERATED BY "LD STAT A HI" + "BD12" H (MAIN ST1).
8774                                        ;* PIN DD ("STP1 OUT") IS WIRED TO PIN BB ("SCHMITT
8775                                        ;* TRIG 2") FOR THIS TEST. "STP1 OUT" PULSES ARE RECEIVED AS
8776                                        ;* "SCHMITT TRIG 2" PULSES WHICH WILL CLEAR CLOCK A'S
8777                                        ;* COUNT REGISTER IF MODE 3 IS SELECTED.
8778                                        ;* IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIC.
8779                                        ;* AND ERROR SWITCH REGISTER OPTIONS ARE USED.
8780                                        ;* AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH
8781                                        ;* THE TEST. SW13=1 WILL INHIBIT THIS FEATURE.
8782                                        ;*
8783                                ;*
8784                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
8785                                ;*
8786                                        ;* YOU MUST WIRE PINS DD AND BB OF J1 TOGETHER.
8787                                        ;*
8788                                        ;* LOGIC TESTS (L + S AT 200) SHOULD BE RUN FIRST.
8789                                        ;*
8790
```

```
 8791  023560                          LS214:
 8792
 8793  023560  005037  001104          1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
 8794  023564  012737  177704  001206          MOV     #-60.,$PASS     ;/SET PASS COUNT.
 8795                                                                  ;/NOTE: PASS COUNT USED ONLY
 8796                                                                  ;/      TO DETECT 60 PASSES SO
 8797                                                                  ;/      IT CAN GENERATE A CRLF.
 8798                                                                  ;/      AFTER CRLF IT WILL BE ZEROED.
 8799  023572  005077  155526          2$:     CLR     @ASR            ;/CLEAR CLOCK A.
 8800  023576  005077  155530                  CLR     @BSR            ;/CLEAR CLOCK B.
 8801
 8802  023602  012777  177777  155516          MOV     #-1,@ABR        ;LOAD ALL ONES TO CLK A'S BUFFER + COUNT REG.
 8803  023610  012777  001400  155506          MOV     #BIT8!BIT9,@ASR ;SELECT MODE 3.
 8804  023616  052777  010000  155500          BIS     #BIT12,@ASR     ;GENERATE A "STP1 OUT" PULSE.
 8805                                                                  ;AT THIS POINT WE SHOULD SEE AN
 8806                                                                  ;OUTPUT AT PIN DD.  PIN DD SHOULD
 8807                                                                  ;BE WIRED TO PIN BB FOR
 8808                                                                  ;"SCHMITT TRIG 2" IN. THIS SHOULD
 8809                                                                  ;CAUSE AN "STP2" WHICH WILL CLEAR
 8810                                                                  ;CLOCK A'S COUNT REGISTER.
 8811  023624  005777  155500                  TST     @ACR            ;WAS THE COUNT REGISTER CLEARED?
 8812
 8813  023630  001401                          BEQ     3$              ;IF YES BR 3$.
 8814
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8818  023632  104000                          ERROR                   ;ERROR "SCHMITT TRIG 2" IN NOT
 8819                                                                  ;RECEIVED. HAVE YOU WIRED IT RIGHT?
 8820
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 8824  023634                          3$:
 8825
 8826  023634  032777  020000  155274          BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
 8827  023642  001353                          BNE     2$              ;/YES - IGNORE ANY UPDATES.
 8828
 8829  023644  005237  001104                  INC     $ICNT           ;/UPDATE COUNT.
 8830  023650  001350                          BNE     2$              ;/IF NOT DONE 65,324 TIMES,
 8831                                                                  ;/DO IT AGAIN.
 8832
 8833  023652  104400  023660                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 8834  023656  000401                          BR      64$             ;;GET OVER THE ASCIZ
 8835                                  ;;65$:  .ASCIZ  #*#
 8836  023662                          64$:
 8837
 8838  023662  005237  001206                  INC     $PASS           ;/DONE 60 PASSES?
 8839  023666  100741                          BMI     2$              ;/NO - NO NEED FOR CR,LF.
 8840  023670  104400  023676                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 8841  023674  000402                          BR      66$             ;;GET OVER THE ASCIZ
 8842                                  ;;67$:  .ASCIZ  <15><12>##
 8843  023702                          66$:
 8844  023702  000726                          BR      1$
 8845
 8846
```

```
                                .SBTTL  *       "SCHMITT TRIG 3" IN, "ST3 OUT" TESTS
                                        ;*
                                        ;*THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND
                                        ;*PROVIDING SCOPE LOOP CAPABILITIES FOR "SCHMITT TRIG 3"
                                        ;*AND "ST3 OUT".
                                        ;*
                                        ;*WHEN YOU LOAD AND START AT LOCATION 220, PROGRAM
                                        ;*CONTROL IS TRANSFERRED HERE. "STP2" PULSES ARE GENERATED
                                        ;*BY "LD STAT A H," + "BD10" H (MAIN STP2). PIN V ("STP2 OUT")
                                        ;*IS WIRED TO PIN T ("SCHMITT TRIG 3"). "SCHMITT TRIG 3" PULSES
                                        ;*GIVE US "ST3 OUT" PULSES. PIN L ("ST3 OUT") IS WIRED
                                        ;*TO PIN LL ("SCHMITT TRIG1"). AND "SCHMITT TRIG 1" WILL SET
                                        ;*CLOCK A'S STATUS REGISTER BIT 15.
                                        ;*IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIC.
                                        ;*AND ERROR SWITCH REGISTER OPTIONS ARE USED.
                                        ;*AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE
                                        ;*TEST. SW13=1 WILL INHIBIT THIS FEATURE.
                                        ;*
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                ;*
                                        ;*YOU MUST WIRE PINS V TO T OF J1 TOGETHER, AS WELL AS
                                        ;*PINS L TO LL OF J1 TOGETHER.
                                        ;*
                                        ;*TESTS LS210 AND LS214 SHOULD BE RUN FIRST.
                                        ;*

023704                          LS220:

023704  005037  001104          1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
023710  012737  177704  001206          MOV     #-60.,$PASS     ;/SET PASS COUNT.
                                                                ;/NOTE: PASS COUNT USED ONLY
                                                                ;/      TO DETECT 60 PASSES SO
                                                                ;/      IT CAN GENERATE A CRLF.
                                                                ;/      AFTER CRLF IT WILL BE ZEROED.
023716  005077  155402          2$:     CLR     @ASR            ;/CLEAR CLOCK A.
023722  005077  155404                  CLR     @BSR            ;/CLEAR CLOCK B.

023726  052777  002000  155370          BIS     #BIT10,@ASR     ;GENERATE A "STP2 OUT" PULSE.
                                                                ;AT THIS POINT YOU SHOULD SEE AN
                                                                ;OUTPUT AT PIN V. PIN V SHOULD BE
                                                                ;WIRED TO PIN T TO GIVE "SCHMITT TRIG3"
                                                                ;INPUT WHICH IN TURN SHOULD GIVE
                                                                ;AN OUTPUT AT PIN L ("ST3"). PIN
                                                                ;L BEING WIRED TO PIN LL ("SCHMITT
                                                                ;TRIG1 IN") SHOULD GIVE "ST1" WHICH
                                                                ;SETS CLOCK A'S CSR BIT 15.

023734  005777  155364                  TST     @ASR            ;DID BIT 15 SET?
023740  100401                          BMI     3$              ;IF YES, BR TO 3$.

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

023742  104000                          ERROR                   ;ERROR "SCHMITT TRIG 1" NOT RECEIVED
                                                                ;HAVE YOU WIRED IT RIGHT?
```

```
                ::;$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$

023744                                  3$:

023744  032777  020000  155164                  BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
023752  001361                                  BNE     2$              ;/YES - IGNORE ANY UPDATES.

023754  005237  001104                          INC     $ICNT           ;/UPDATE COUNT.
023760  001356                                  BNE     2$              ;/IF NOT DONE 65,324 TIMES,
                                                                        ;/DO IT AGAIN.

023762  104400  023770                          TYPE    ,65$            ;;TYPE ASCIZ STRING
023766  000401                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  ***
023772                                  64$:

023772  005237  001206                          INC     $PASS           ;/DONE 60 PASSES?
023776  100747                                  BMI     2$              ;/NO - NO NEED FOR CR,LF.
024000  104400  024006                          TYPE    ,67$            ;;TYPE ASCIZ STRING
024004  000402                                  BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>**
024012                                  66$:
024012  000734                                  BR      1$


                                        .SBTTL  *       "A EVENT OUT" TEST
                                                ;*
                                                ;*THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND
                                                ;*PROVIDING SCOPE LOOP CAPABILITIES FOR "A EVENT OUT".
                                                ;*
                                                ;*WHEN YOU LOAD AND START AT LOCATION 224, PROGRAM
                                                ;*CONTROL IS TRANSFERRED HERE. "A EVENT OUT" PULSES ARE
                                                ;*GENERATED BY CLOCK A OVERFLOWS. PIN VV
                                                ;*("A EVENT OUT") IS WIRED TO PIN LL ("SCHMITT TRIG 1")
                                                ;*"SCHMITT TRIG 1" PULSES WILL SET CLOCK A'S CSR BIT 15.
                                                ;*IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIQ.
                                                ;*AND ERROR SWITCH REGISTER OPTIONS ARE USED.
                                                ;*AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE TEST.
                                                ;*SW13=1 WILL INHIBIT THIS FEATURE.
                                                ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
                                        ;*
                                                ;*YOU MUST WIRE PINS VV AND LL OF J1 TOGETHER.
                                                ;*
                                                ;*TEST LS210 SHOULD BE RUN FIRST.
                                                ;*

024014                                  LS224:

024014  005037  001104                  1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
024020  012737  177704  001206                  MOV     #-60.,$PASS     ;/SET PASS COUNT.
                                                                        ;/NOTE: PASS COUNT USED ONLY
                                                                        ;/      TO DETECT 60 PASSES SO
```

```
8959
8960
8961  024026  005077  155272          2$:   CLR   @ASR          ;/      IT CAN GENERATE A CRLF.
8962  024032  005077  155274                CLR   @BSR          ;/      AFTER CRLF IT WILL BE ZEROED.
                                                                ;/CLEAR CLOCK A.
                                                                ;/CLEAR CLOCK B.
8963
8964  024036  012777  177777  155262        MOV   #-1,@ABR     ;PRELOAD CLOCK A'S BUFFER + COUNT REGS.
8965
8966  024044  012777  000003  155252        MOV   #3,@ASR      ;RATE: 1MHZ, ENABLE COUNTER
8967
8968  024052  032777  000040  155244  3$:   BIT   #BIT05,@ASR  ;HAS AN OVERFLOW OCCURRED?
8969  024060  001774                BEQ   3$                    ;NO - THEN WAIT FOR IT.
8970                                                            ;OVERFLOW SHOULD GO OUT AS
8971                                                            ;"A EVENT OUT". YOU SHOULD SEE
8972                                                            ;AN OUTPUT AT PIN VV.
8973                                                            ;THIS IN TURN IS BROUGHT
8974                                                            ;IN AS "SCHMITT TRIG 1" IN TO
8975                                                            ;SET CLOCK A'S CSR BIT 15.
8976
8977  024062  005777  155236                TST   @ASR          ;DID CSR BIT 15 SET?
8978  024066  100401                BMI   4$                    ;BR IF YES TO 4$
8979

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8983  024070  104000                ERROR                       ;ERROR "A EVENT OUT" PULSE
8984                                                            ;NOT DETECTED. HAVE YOU
8985                                                            ;WIRED IT RIGHT?
8986

        ;;;SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR <<SSSSSSSSSSSSSSSSSSSSSSSS

8990  024072                  4$:
8991
8992  024072  032777  020000  155036        BIT   #BIT13,@SWR   ;/INHIBIT "*" TYPEOUT?
8993  024100  001352                BNE   2$                    ;/YES - IGNORE ANY UPDATES.
8994
8995  024102  005237  001104        INC   $ICNT                 ;/UPDATE COUNT.
8996  024106  001347                BNE   2$                    ;/IF NOT DONE 65,324 TIMES,
8997                                                            ;/DO IT AGAIN.
8998
8999  024110  104400  024116        TYPE  ,65$                  ;;TYPE ASCIZ STRING
9000  024114  000401                BR    64$                   ;;GET OVER THE ASCIZ
9001                          ;;65$:  .ASCIZ  **
9002  024120                  64$:
9003
9004  024120  005237  001206        INC   $PASS                 ;/DONE 60 PASSES?
9005  024124  100740                BMI   2$                    ;/NO - NO NEED FOR CR,LF.
9006  024126  104400  024134        TYPE  ,67$                  ;;TYPE ASCIZ STRING
9007  024132  000402                BR    66$                   ;;GET OVER THE ASCIZ
9008                          ;;67$:  .ASCIZ  <15><12>**
9009  024140                  66$:
9010  024140  000725                BR    1$
9011
9012
9013                          .SBTTL  *       "B EVENT OUT" TEST
9014                                  ;*
```

```
                                        ;*THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND
                                        ;*PROVIDING SCOPE LOOP CAPABILITIES FOR "B EVENT OUT".
                                        ;*
                                        ;*WHEN YOU LOAD AND START AT LOCATION 230, PROGRAM
                                        ;*CONTROL IS TRANSFERRED HERE. "B EVENT OUT" PULSES ARE
                                        ;*GENERATED BY CLOCK B OVERFLOWS. PIN TT
                                        ;*("B EVENT OUT") IS WIRED TO PIN LL ("SCHMITT TRIG 1")
                                        ;*"SCHMITT TRIG 1" PULSES WILL SET CLOCK A'S CSR BIT 15.
                                        ;*IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIQ.
                                        ;*AND ERROR SWITCH REGISTER OPTIONS ARE USED.
                                        ;*AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE TEST.
                                        ;*SW13=1 WILL INHIBIT THIS FEATURE.
                                        ;*
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                ;*
                                        ;*YOU MUST WIRE PINS TT AND LL OF J1 TOGETHER.
                                        ;*
                                        ;*TEST LS210 SHOULD BE RUN FIRST.
                                        ;*

024142                          LS230:

024142  005037  001104          1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
024146  012737  177704  001206          MOV     #-60.,$PASS     ;/SET PASS COUNT.
                                                                ;/NOTE: PASS COUNT USED ONLY
                                                                ;/      TO DETECT 60 PASSES SO
                                                                ;/      IT CAN GENERATE A CRLF.
                                                                ;/      AFTER CRLF IT WILL BE ZEROED.
024154  005077  155144          2$:     CLR     @ASR            ;/CLEAR CLOCK A.
024160  005077  155146                  CLR     @BSR            ;/CLEAR CLOCK B.

024164  012777  177777  155142          MOV     #-1,@BBR        ;PRELOAD CLOCK B'S BUFFER + COUNT REGS.

024172  012777  000003  155132          MOV     #3,@BSR         ;RATE: 1MHZ, ENABLE COUNTER

024200  032777  000200  155124  3$:     BIT     #BIT07,@BSR     ;HAS AN OVERFLOW OCCURRED?
024206  001774                          BEQ     3$              ;NO - THEN WAIT FOR IT.
                                                                ;OVERFLOW SHOULD GO OUT AS
                                                                ;"B EVENT OUT". YOU SHOULD SEE
                                                                ;AN OUTPUT AT PIN TT.
                                                                ;THIS IN TURN IS BROUGHT
                                                                ;IN AS "SCHMITT TRIG 1" IN TO
                                                                ;SET CLOCK A'S CSR BIT 15.

024210  005777  155110                  TST     @ASR            ;DID CSR BIT 15 SET?
024214  100401                          BMI     4$              ;BR IF YES TO 4$

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

024216  104000                          ERROR                   ;ERROR "B EVENT OUT" PULSE
                                                                ;NOT DETECTED. HAVE YOU
                                                                ;WIRED IT RIGHT?

        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
9073  024220                                  4$:
9074
9075  024220  032777  020000  154710          BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
9076  024226  001352                          BNE     2$              ;/YES - IGNORE ANY UPDATES.
9077
9078  024230  005237  001104                  INC     $ICNT           ;/UPDATE COUNT.
9079  024234  001347                          BNE     2$              ;/IF NOT DONE 65,324 TIMES,
9080                                                                  ;/DO IT AGAIN.
9081
9082  024236  104400  024244                  TYPE    ,65$            ;;TYPE ASCIZ STRING
9083  024242  000401                          BR      64$             ;;GET OVER THE ASCIZ
9084                                  ;;65$:  .ASCIZ  #*#
9085  024246                          64$:
9086
9087  024246  005237  001206                  INC     $PASS           ;/DONE 60 PASSES?
9088  024252  100740                          BMI     2$              ;/NO - NO NEED FOR CR,LF.
9089  024254  104400  024262                  TYPE    ,67$            ;;TYPE ASCIZ STRING
9090  024260  000402                          BR      66$             ;;GET OVER THE ASCIZ
9091                                  ;;67$:  .ASCIZ  <15><12>#*
9092  024266                          66$:
9093  024266  000725                          BR      1$
9094
9095
9096                                  ;*ROUTINE TO HANDLE TRAPS TO LOC 4, 10 AND .
9097                                  ;*INTERRUPTS TO WRONG VECTORS.
9098                                  ;*.+2, IOTT(TRAPS) WERE PUT IN LOCATIONS 4-1000
9099
9100
9101  024270                          IOTRD:
9102  024270  011637  024440                  MOV     (R6),2$         ;GET WHERE WE TRAPPED TO.
9103  024274  162737  000004  024440          SUB     #4,2$           ;=WHERE R6 RETURN 10-4
9104  024302  104400  024310                  TYPE    ,65$            ;;TYPE ASCIZ STRING:
9105  024306  000412                          BR      64$             ;;GET OVER THE ASCIZ
9106                                  ;;65$:  .ASCIZ  <15><12>#ILLEGAL TRAP TO: #
9107  024334                          64$:
9108
9109  024334  013746  024440                  MOV     2$,-(SP)        ;;SAVE 2$ FOR TYPEOUT
9110  024340  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
9111
9112  024342  104400  024350                  TYPE    ,67$            ;;TYPE ASCIZ STRING
9113  024346  000407                          BR      66$             ;;GET OVER THE ASCIZ
9114                                  ;;67$:  .ASCIZ  # FROM LOC.: #
9115  024366                          66$:
9116
9117  024366  062706  000004                  ADD     #4,R6           ;POINT TO WHERE WE TRAPPED FROM.
9118
9119  024372  011637  024442                  MOV     (R6),3$         ;PICK UP LOC
9120  024376  162737  000002  024442          SUB     #2,3$           ;FROM REAL ADDR.
9121  024404  013746  024442                  MOV     3$,-(SP)        ;;SAVE 3$ FOR TYPEOUT
9122  024410  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
9123
9124  024412  023727  024440  000004          CMP     2$,#4           ;DID WE TRAP TO LOC 4?
9125  024420  001405                          BEQ     1$              ;IF SO - DON'T RETURN!
9126  024422  023727  024440  000010          CMP     2$,#10          ;DID WE TRAP TO LOC. 10?
```

```
 9127  024430  001401                          BEQ     1$              ;ID SO - DON'T RETURN!
 9128  024432  000002                          RTI                     ;TRY RETURNING.
 9129
 9130
                ;;;$$$
```

```
 9183  024512  010346                        MOV     R3,-(SP)        ;;SAVE R3
 9184  024514  010446                        MOV     R4,-(SP)        ;;SAVE R4
 9185  024516  010546                        MOV     R5,-(SP)        ;;SAVE R5
 9186  024520  113704  024671                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 9187  024524  005404                        NEG     R4
 9188  024526  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 9189  024532  110437  024670                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 9190  024536  113704  024667                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 9191  024542  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 9192  024546  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
 9193  024550  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 9194  024552  000404                        BR      3$              ;;GO DO MSB
 9195  024554  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
 9196  024556  006105                        ROL     R5
 9197  024560  006105                        ROL     R5
 9198  024562  010503                        MOV     R5,R3
 9199  024564  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 9200  024566  105337  024670                DECB    $OMODE          ;;TYPE THIS DIGIT?
 9201  024572  100016                        BPL     7$              ;;BR IF NO
 9202  024574  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
 9203  024600  001002                        BNE     4$              ;;TEST FOR 0
 9204  024602  005704                        TST     R4              ;;SUPPRESS THIS 0?
 9205  024604  001403                        BEQ     5$              ;;BR IF YES
 9206  024606  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 9207  024610  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 9208  024614  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 9209  024620  110337  024664                MOVB    R3,8$           ;;SAVE FOR TYPING
 9210  024624  104400  024664                TYPE    ,8$             ;;GO TYPE THIS DIGIT
 9211  024630  105337  024666        7$:     DECB    $OCNT           ;;COUNT BY 1
 9212  024634  003347                        BGT     2$              ;;BR IF MORE TO DO
 9213  024636  002402                        BLT     6$              ;;BR IF DONE
 9214  024640  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 9215  024642  000744                        BR      2$              ;;GO DO THE LAST DIGIT
 9216  024644  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
 9217  024646  012604                        MOV     (SP)+,R4        ;;RESTORE R4
 9218  024650  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 9219  024652  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 9220  024660  012616                        MOV     (SP)+,(SP)
 9221  024662  000002                        RTI                     ;;RETURN
 9222  024664     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 9223  024665     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 9224  024666     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 9225  024667     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 9226  024670  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 9227
 9228                                .SBTTL  ERROR HANDLER ROUTINE
 9229
 9230                                ;;**************************************************************
 9231                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 9232                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 9233                                ;*AND GO TO SERRTYP ON ERROR
 9234                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 9235                                ;*SW15=1         HALT ON ERROR
 9236                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
 9237                                ;*SW10=1         BELL ON ERROR
 9238                                ;*SW09=1         LOOP ON ERROR
```

```
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

024672                                  $ERROR:
024672  105237  001103                7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
024676  001775                              BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
024700  013777  001102  154232              MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
024706  032777  002000  154222              BIT     #BIT10,@SWR     ;;BELL ON ERROR?
024714  001402                              BEQ     1$              ;;NO - SKIP
024716  104400  001170                      TYPE    ,$BELL          ;;RING BELL
024722  005237  001112                1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
024726  011637  001116                      MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
024732  162737  000002  001116              SUB     #2,$ERRPC
024740  117737  154152  001114              MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
024746  032777  020000  154162              BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
024754  001004                              BNE     20$             ;;SKIP TYPEOUTS
024756  004737  025054                      JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
024762  104400  001175                      TYPE    ,$CRLF
024766                                  20$:
024766  122737  000001  001220              CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
024774  001007                              BNE     2$              ;;NO,SKIP APT ERROR REPORT
024776  113737  001114  025010              MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
025004  004737  026704                      JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
025010     000                          21$:    .BYTE   0
025011     000                              .BYTE   0
025012  000777                          22$:    BR      22$             ;;APT ERROR LOOP
025014  005777  154116                2$:     TST     @SWR            ;;HALT ON ERROR
025020  100001                              BPL     3$              ;;SKIP IF CONTINUE
025022  000000                              HALT                    ;;HALT ON ERROR!
025024  032777  001000  154104          3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
025032  001402                              BEQ     4$              ;;BR IF NO
025034  013716  001110                      MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
025040  005737  001166                4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
025044  001402                              BEQ     5$              ;;BR IF NONE
025046  013716  001166                      MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
025052                                  5$:
025052  000002                              RTI                     ;;RETURN

                                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

025054                                  $ERRTYP:
025054  104400  001175                      TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025060  010046                              MOV     R0,-(SP)        ;;SAVE R0
025062  005000                              CLR     R0              ;;PICKUP THE ITEM INDEX
025064  153700  001114                      BISB    @#$ITEMB,R0
025070  001004                              BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                    ;;TYPE THE PC OF THE ERROR
025072  013746  001116                      MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                    ;;ERROR ADDRESS
025076  104401                              TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025100  000426                              BR      6$              ;;GET OUT
```

```
 9295  025102  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 9296  025104  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
 9297  025106  006300                          ASL     R0
 9298  025110  006300                          ASL     R0
 9299  025112  062700  001354                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 9300  025116  012037  025126                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
 9301  025122  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 9302  025124  104400                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
 9303  025126  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
 9304  025130  104400  001175                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 9305  025134  012037  025144          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
 9306  025140  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
 9307  025142  104400                          TYPE                    ;;TYPE THE "DATA HEADER"
 9308  025144  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
 9309  025146  104400  001175                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 9310  025152  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
 9311  025154  001004                          BNE     7$              ;;GO TYPE THE DATA
 9312  025156  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
 9313  025160  104400  001175                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 9314  025164  000207                          RTS     PC              ;;RETURN
 9315  025166                          7$:
 9316  025166  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
 9317  025170  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9318  025172  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
 9319  025174  001770                          BEQ     6$              ;;BR IF NO
 9320  025176  104400  025204                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
 9321  025202  000771                          BR      7$              ;;LOOP
 9322  025204  020040      000         8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
 9323          025210                          .EVEN
 9324
 9325                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 9326
 9327                                  ;;**************************************************************
 9328                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 9329                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 9330                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 9331                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 9332                                  ;*REPLACED WITH SPACES.
 9333                                  ;*CALL:
 9334                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 9335                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
 9336
 9337  025210                          $TYPDS:
 9338  025210  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9339  025212  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9340  025214  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9341  025216  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9342  025220  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9343  025222  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 9344  025226  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 9345  025232  100004                          BPL     1$              ;;BR IF INPUT IS POS.
 9346  025234  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 9347  025236  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 9348  025244  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 9349  025246  012703  025424                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 9350  025252  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
```

```
 9351  025256  005002                  2$:   CLR     R2              ;;CLEAR THE BCD NUMBER
 9352  025260  016001  025414                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 9353  025264  160105                  3$:   SUB     R1,R5           ;;FORM THIS BCD DIGIT
 9354  025266  002402                        BLT     4$              ;;BR IF DONE
 9355  025270  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 9356  025272  000774                        BR      3$
 9357  025274  060105                  4$:   ADD     R1,R5           ;;ADD BACK THE CONSTANT
 9358  025276  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
 9359  025300  001002                        BNE     5$              ;;FALL THROUGH IF 0
 9360  025302  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 9361  025304  100407                        BMI     7$              ;;BR IF YES
 9362  025306  106316                  5$:   ASLB    (SP)            ;;MSD?
 9363  025310  103003                        BCC     6$              ;;BR IF NO
 9364  025312  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 9365  025320  052702  000060          6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 9366  025324  052702  000040          7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 9367  025330  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 9368  025332  005720                        TST     (R0)+           ;;JUST INCREMENTING
 9369  025334  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
 9370  025340  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
 9371  025342  003002                        BGT     8$              ;;GO TO EXIT
 9372  025344  010502                        MOV     R5,R2           ;;GET THE LSD
 9373  025346  000764                        BR      6$              ;;GO CHANGE TO ASCII
 9374  025350  105726                  8$:   TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 9375  025352  100003                        BPL     9$              ;;BR IF NO
 9376  025354  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 9377  025362  105013                  9$:   CLRB    (R3)            ;;SET THE TERMINATOR
 9378  025364  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 9379  025366  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 9380  025370  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9381  025372  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9382  025374  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9383  025376  104400  025424                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 9384  025402  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 9385  025410  012616                        MOV     (SP)+,(SP)
 9386  025412  000002                        RTI                     ;;RETURN TO USER
 9387  025414  023420                  $DTBL:  10000.
 9388  025416  001750                        1000.
 9389  025420  000144                        100.
 9390  025422  000012                        10.
 9391  025424  000004                  $DBLK:  .BLKW   4
 9392
 9393                                   .SBTTL  SCOPE HANDLER ROUTINE
 9394
 9395                                   ;;**********************************************************************
 9396                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 9397                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 9398                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 9399                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 9400                                   ;*SW14=1         LOOP ON TEST
 9401                                   ;*SW11=1         INHIBIT ITERATIONS
 9402                                   ;*SW09=1         LOOP ON ERROR
 9403                                   ;*SW08=1         LOOP ON TEST IN SWR<7:0>
 9404                                   ;*CALL
 9405                                   ;*      SCOPE           ;;SCOPE=IOT
 9406
```

```
 9407  025434                          $SCOPE:
 9408  025434  032777  040000  153474  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 9409  025442  001114                          BNE     $OVER           ;;YES IF SW14=1
 9410                                  ;*****START OF CODE FOR THE XOR TESTER*****
 9411  025444  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 9412                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 9413  025446  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 9414  025452  012737  025472  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 9415  025460  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
 9416  025464  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9417  025470  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
 9418  025472  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 9419  025474  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9420  025500  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
 9421  025502                          6$:;*****END OF CODE FOR THE XOR TESTER*****
 9422  025502  032777  000400  153426          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
 9423  025510  001404                          BEQ     2$              ;;BR IF NO
 9424  025512  127737  153420  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
 9425  025520  001465                          BEQ     $OVER           ;;BR IF YES
 9426  025522  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 9427  025526  001421                          BEQ     3$              ;;BR IF NO
 9428  025530  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 9429  025536  101015                          BHI     3$              ;;BR IF NO
 9430  025540  032777  001000  153370          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 9431  025546  001404                          BEQ     4$              ;;BR IF NO
 9432  025550  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 9433  025556  000446                          BR      $OVER
 9434  025560  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 9435  025564  005037  001164                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 9436  025570  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
 9437  025572  032777  004000  153336  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 9438  025600  001011                          BNE     1$              ;;BR IF YES
 9439  025602  005737  001206                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 9440  025606  001406                          BEQ     1$              ;;     INHIBIT ITERATIONS
 9441  025610  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
 9442  025614  023737  001164  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 9443  025622  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 9444  025624  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 9445  025632  013737  025710  001164          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 9446  025640  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 9447  025644  113737  001102  001204          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
 9448  025652  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 9449  025656  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 9450  025662  005037  001166                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 9451  025666  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 9452  025674  013777  001102  153236  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 9453  025702  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 9454  025706  000002                          RTI                     ;;FIXES PS
 9455  025710  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 9456
 9457                                  .SBTTL  TTY INPUT ROUTINE
 9458
 9459                                  ;;**************************************************************
 9460                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 9461                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
 9462                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
```

```
                                ;*WHEN OPERATING IN TTY FLAG MODE.
025712 022737 000176 001136 $CKSWR: CMP   #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
025720 001073                       BNE   14$             ;;BRANCH IF NO
025722 105777 153214                TSTB  @$TKS           ;;CHAR THERE?
025726 100070                       BPL   14$             ;;IF NO, DON'T WAIT AROUND
025730 117746 153210          2$:   MOVB  @$TKB,-(SP)     ;;SAVE THE CHAR
025734 042716 177600                BIC   #^C177,(SP)     ;;STRIP-OFF THE ASCII
025740 022726 000007                CMP   #7,(SP)+        ;;IS IT A CONTROL G?
025744 001061                       BNE   14$             ;;NO, RETURN TO USER
025746 104400 026355                TYPE  ,$CNTLG         ;;YES, ECHO CONTROL G

025752 104400 026362          6$:   TYPE  ,$MSWR          ;;TYPE CURRENT CONTENTS
025756 013746 000176                MOV   $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
025762 104401                       TYPOC                 ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025764 104400 026373                TYPE  ,$MNEW          ;;PROMPT FOR NEW SWR
025770 005046                       CLR   -(SP)           ;;CLEAR COUNTER
025772 005046                       CLR   -(SP)           ;;THE NEW SWR
025774 104406                 7$:   RDCHR                 ;;GET NEXT CHAR

025776 022716 000025          8$:   CMP   #25,(SP)        ;;IS IT A CONTROL U?
026002 001005                       BNE   9$              ;;BRANCH IF NO
026004 104400 026350                TYPE  ,$CNTLU         ;;YES, ECHO IT
026010 062706 000006                ADD   #6,SP           ;;IGNORE PREVIOUS INPUT
026014 000756                       BR    6$              ;;LET'S TRY IT AGAIN

026016 022716 000015          9$:   CMP   #15,(SP)        ;;IS IT A <CR>?
026022 001011                       BNE   11$             ;;BRANCH IF NO
026024 005766 000004                TST   4(SP)           ;;YES, IS IT THE FIRST CHAR?
026030 001403                       BEQ   10$             ;;BRANCH IF YES
026032 016677 000002 153076         MOV   2(SP),@SWR      ;;SAVE NEW SWR
026040 062706 000006          10$:  ADD   #6,SP           ;;CLEAR UP STACK
026044 000417                       BR    13$             ;;RETURN TO USER
026046 022716 000012          11$:  CMP   #12,(SP)        ;;IS IT A <LF>
026052 001017                       BNE   15$             ;;BRANCH IF NO
026054 005766 000004                TST   4(SP)           ;;YES, IS IT THE FIRST CHAR?
026060 001403                       BEQ   12$             ;;YES
026062 016677 000002 153046         MOV   2(SP),@SWR      ;;SAVE NEW SWR
026070 062706 000006          12$:  ADD   #6,SP           ;;CLEAR UP STACK
026074 013716 000046                MOV   @#46,(SP)       ;;GET RESTART
026100 062716 000010                ADD   #10,(SP)        ;;ADDRESS
026104 104400 001175          13$:  TYPE  ,$CRLF          ;;ECHO <CR> AND <LF>
026110 000002                 14$:  RTI                   ;;RETURN
026112 004737 026616          15$:  JSR   PC,$TYPEC       ;;ECHO CHAR
026116 042726 177770                BIC   #177770,(SP)+   ;;RESTRICT TO 0-7
026122 005766 000002                TST   2(SP)           ;;IS THIS THE FIRST CHAR
026126 001403                       BEQ   16$             ;;BRANCH IF YES
026130 006316                       ASL   (SP)            ;;NO, SHIFT PRESENT
026132 006316                       ASL   (SP)            ;;   CHAR OVER TO MAKE
026134 006316                       ASL   (SP)            ;;   ROOM FOR NEW ONE.
026136 005266 000002          16$:  INC   2(SP)           ;;KEEP COUNT OF CHAR
026142 056616 177776                BIS   -2(SP),(SP)     ;;SET IN NEW CHAR
026146 000712                       BR    7$              ;;GET THE NEXT ONE
                                ;;**************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
```

```
                                        ;*      RETURN HERE                 ;;CHARACTER IS ON THE STACK
                                        ;*                                  ;;WITH PARITY BIT STRIPPED OFF
                                        ;
026150  011646                          $RDCHR: MOV     (SP),-(SP)          ;;PUSH DOWN THE PC
026152  016666  000004  000002                  MOV     4(SP),2(SP)         ;;SAVE THE PS
026160  105777  152756                  1$:     TSTB    @$TKS               ;;WAIT FOR
026164  100375                                  BPL     1$                  ;;A CHARACTER
026166  117766  152752  000004                  MOVB    @$TKB,4(SP)         ;;READ THE TTY
026174  042766  177600  000004                  BIC     #↑C<177>,4(SP)      ;;GET RID OF JUNK IF ANY
026202  026627  000004  000140                  CMP     4(SP),#140          ;;IS IT UPPER CASE?
026210  002407                                  BLT     2$                  ;;BRANCH IF YES
026212  026627  000004  000175                  CMP     4(SP),#175          ;;IS IT A SPECIAL CHAR?
026220  003003                                  BGT     2$                  ;;BRANCH IF YES
026222  042766  000040  000004                  BIC     #40,4(SP)           ;;MAKE IT UPPER CASE
026230  000002                          2$:     RTI                         ;;GO BACK TO USER
                                        ;;*******************************************************************
                                        ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                        ;*CALL:
                                        ;*      RDLIN                       ;;INPUT A STRING FROM THE TTY
                                        ;*      RETURN HERE                 ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                                  ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
026232  010346                          $RDLIN: MOV     R3,-(SP)            ;;SAVE R3
026234  012703  026340                  1$:     MOV     #$TTYIN,R3          ;;GET ADDRESS
026240  022703  026350                  2$:     CMP     #$TTYIN+8.,R3       ;;BUFFER FULL?
026244  101405                                  BLOS    4$                  ;;BR IF YES
026246  104406                                  RDCHR                       ;;GO READ ONE CHARACTER FROM THE TTY
026250  112613                                  MOVB    (SP)+,(R3)          ;;GET CHARACTER
026252  122713  000177                  10$:    CMPB    #177,(R3)           ;;IS IT A RUBOUT
026256  001003                                  BNE     3$                  ;;SKIP IF NOT
026260  104400  001174                  4$:     TYPE    ,$QUES              ;;TYPE A '?'
026264  000763                                  BR      1$                  ;;CLEAR THE BUFFER AND LOOP
026266  111337  026336                  3$:     MOVB    (R3),9$             ;;ECHO THE CHARACTER
026272  104400  026336                          TYPE    ,9$
026276  122723  000015                          CMPB    #15,(R3)+           ;;CHECK FOR RETURN
026302  001356                                  BNE     2$                  ;;LOOP IF NOT RETURN
026304  105063  177777                          CLRB    -1(R3)              ;;CLEAR RETURN (THE 15)
026310  104400  001176                          TYPE    ,$LF                ;;TYPE A LINE FEED
026314  012603                                  MOV     (SP)+,R3            ;;RESTORE R3
026316  011646                                  MOV     (SP),-(SP)          ;;ADJUST THE STACK AND PUT ADDRESS OF THE
026320  016666  000004  000002                  MOV     4(SP),2(SP)         ;;      FIRST ASCII CHARACTER ON IT
026326  012766  026340  000004                  MOV     #$TTYIN,4(SP)
026334  000002                                  RTI                         ;;RETURN
026336     000                          9$:     .BYTE   0                   ;;STORAGE FOR ASCII CHAR. TO TYPE
026337     000                                  .BYTE   0                   ;;TERMINATOR
026340  000010                          $TTYIN: .BLKB   8.                  ;;RESERVE 8 BYTES FOR TTY INPUT
026350  052536  005015     000          $CNTLU: .ASCIZ  /↑U/<15><12>        ;;CONTROL "U"
026355     136  006507  000012          $CNTLG: .ASCIZ  /↑G/<15><12>        ;;CONTROL "G"
026362  005015  053523  020122          $MSWR:  .ASCIZ  <15><12>/SWR = /
026370  020075     000
026373     040  047040  053505          $MNEW:  .ASCIZ  /  NEW = /
026400  036440  000040

                                        .SBTTL  TYPE ROUTINE
```

```
                                ;;**********************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*       TYPE
                                ;*       MESADR
                                ;*

026404  105737  001155          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
026410  100002                          BPL     1$              ;;BR IF YES
026412  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
026414  000430                          BR      3$              ;;LEAVE
026416  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
026420  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
026424  122737  000001  001220          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
026432  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
026434  132737  000100  001221          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
026442  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
026444  010037  026454                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
026450  004737  026674                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
026454  000000          61$:    .WORD   0                       ;;MESSAGE ADDRESS
026456  132737  000040  001221  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
026464  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
026466  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
026470  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
026472  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
026474  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
026476  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
026502  000002                          RTI                     ;;RETURN
026504  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
026510  001430                          BEQ     8$
026512  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
026516  001006                          BNE     5$
026520  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
026522  104400                          TYPE                    ;;TYPE A CR AND LF
026524  001175                          $CRLF
026526  105037  026662                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
026532  000755                          BR      2$              ;;GET NEXT CHARACTER
026534  004737  026616          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
026540  123726  001154          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
026544  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
026546  013746  001152                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
026552  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
026556  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
026560  004737  026616                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
026564  105337  026662                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
026570  000770                          BR      7$              ;;LOOP
```

```
                                        ;HORIZONTAL TAB PROCESSOR

026572  112716  000040          B$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
026576  004737  026616          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
026602  132737  000007  026662          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
026610  001372                          BNE     9$              ;;TAB STOP
026612  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
026614  000724                          BR      2$              ;;GET NEXT CHARACTER
026616  105777  152324          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
026622  100375                          BPL     $TYPEC
026624  116677  000002  152316          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
026632  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
026640  001003                          BNE     1$              ;;BRANCH IF NO
026642  105037  026662                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
026646  000406                          BR      $TYPEX          ;;EXIT
026650  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
026656  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
026660  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
026662  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
026664  000207          $TYPEX: RTS     PC

                                        .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;**************************************************************
026666  112737  000001  027132  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
026674  112737  000001  027130  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
026702  000403                          BR      $ATYC
026704  112737  000001  027132  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
026712                          $ATYC:
026712  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
026714  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
026716  105737  027130                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
026722  001450                          BEQ     5$              ;;IF NOT:  BR
026724  122737  000001  001220          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
026732  001031                          BNE     3$              ;;IF NOT:  BR
026734  132737  000100  001221          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
026742  001425                          BEQ     3$              ;;IF NOT:  BR
026744  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
026750  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
026756  005737  001200          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
026762  001375                          BNE     1$              ;;IF NOT:  WAIT
026764  010037  001214                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
026770  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
026772  001376                          BNE     2$
026774  163700  001214                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
027000  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
027002  010037  001216                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
027006  012737  000004  001200          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
027014  000413                          BR      5$
027016  017637  000004  027042  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
027024  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
027032  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
027036  004737  026404                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
027042  000000                  4$:     .WORD   0
027044                          5$:
```

```
 9687  027044  105737  027132          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9688  027050  001416                          BEQ     12$             ;;IF NOT:  BR
 9689  027052  005737  001220                  TST     $ENV            ;;RUNNING UNDER APT?
 9690  027056  001413                          BEQ     12$             ;;IF NOT:  BR
 9691  027060  005737  001200          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9692  027064  001375                          BNE     11$             ;;IF NOT:  WAIT
 9693  027066  017637  000004  001202          MOV     @4(SP),$FATAL   ;;GET ERROR #
 9694  027074  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9695  027102  005237  001200                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9696  027106  105037  027132          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 9697  027112  105037  027131                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 9698  027116  105037  027130                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9699  027122  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9700  027124  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9701  027126  000207                          RTS     PC              ;;RETURN
 9702  027130     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
 9703  027131     000          $LFLG:  .BYTE   0               ;;LOG FLAG
 9704  027132     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
 9705          027134                  .EVEN
 9706          000200          APTSIZE=200
 9707          000001          APTENV=001
 9708          000100          APTSPOOL=100
 9709          000040          APTCSUP=040
 9710
 9711                          .SBTTL  POWER DOWN AND UP ROUTINES
 9712
 9713                          ;;**************************************************************
 9714                          ;POWER DOWN ROUTINE
 9715  027134  012737  027274  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 9716  027142  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
 9717  027150  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9718  027152  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9719  027154  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9720  027156  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9721  027160  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 9722  027162  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9723  027164  017746  151746                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
 9724  027170  010637  027300                  MOV     SP,$SAVR6       ;;SAVE SP
 9725  027174  012737  027206  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 9726  027202  000000                          HALT
 9727  027204  000776                          BR      .-2             ;;HANG UP
 9728
 9729
 9730                          ;;**************************************************************
 9731  027206  012737  027274  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 9732  027214  013706  027300                  MOV     $SAVR6,SP       ;;GET SP
 9733  027220  005037  027300                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 9734  027224  005237  027300          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 9735  027230  001375                          BNE     1$              ;;OF  WORD
 9736  027232  012677  151700                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
 9737  027236  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 9738  027240  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 9739  027242  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 9740  027244  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 9741  027246  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9742  027250  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
```

```
 9743  027252  012737  027134  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 9744  027260  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
 9745  027266  104400                          TYPE                    ;REPORT THE POWER FAILURE
 9746  027270  027302                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
 9747  027272  000002                          RTI
 9748  027274  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
 9749  027276  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 9750  027300  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
 9751  027302  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
 9752  027310  000122
 9753                                          .EVEN
 9754
 9755                                  .SBTTL  TRAP DECODER
 9756
 9757                                  ;;**************************************************************
 9758                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 9759                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 9760                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 9761                                  ;*GO TO THAT ROUTINE.
 9762
 9763  027312  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 9764  027314  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
 9765  027320  005740                          TST     -(R0)           ;;BACKUP BY 2
 9766  027322  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 9767  027324  006300                          ASL     R0              ;;POSITION FOR INDEXING
 9768  027326  016000  027334                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 9769  027332  000200                          RTS     R0              ;;GO TO ROUTINE
 9770
 9771
 9772                                  .SBTTL  TRAP TABLE
 9773
 9774                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 9775                                  ;*BY THE "TRAP" INSTRUCTION.
 9776
 9777                                  ;       ROUTINE
 9778                                  ;       -------
 9779  027334                  $TRPAD:
 9780  027334  026404                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
 9781  027336  024470                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 9782  027340  024444                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 9783  027342  024504                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 9784  027344  025210                          $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
 9785  027346  025712                          $CKSWR  ;;CALL=CKSWR    TRAP+5(104405)  TEST FOR CHANGE IN SOFT-SWR
 9786  027350  026150                          $RDCHR  ;;CALL=RDCHR    TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
 9787  027352  026232                          $RDLIN  ;;CALL=RDLIN    TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
 9788  027354  024270                          IOTRD   ;;CALL=IOTT     TRAP+10(104410)
 9789  027356  005015  046103  041517  EM1:    .ASCIZ  <15><12>/CLOCKA SR FUNCTION ERROR/
 9790  027364  040513  051440  020122
 9791  027372  052506  041516  044524
 9792  027400  047117  042440  051122
 9793  027406  051117     000
 9794  027411     015  041412  047514  EM2:    .ASCIZ  <15><12>/CLOCKA SR DATA ERROR/
 9795  027416  045503  020101  051123
 9796  027424  042040  052101  020101
 9797  027432  051105  047522  000122
 9798  027440  005015  046103  041517  EM3:    .ASCIZ  <15><12>/CLOCKA BR DATA ERROR/
```

```
9799  027446  040513  041040  020122
9800  027454  040504  040524  042440
9801  027462  051122  051117     000
9802  027467     015  041412  047514  EM4:    .ASCIZ  <15><12>/CLOCKA CR DATA ERROR/
9803  027474  045503  020101  051103
9804  027502  042040  052101  020101
9805  027510  051105  047522  000122
9806  027516  005015  046103  041517  EM5:    .ASCIZ  <15><12>/CLOCKB SR DATA EPROR/
9807  027524  041113  051440  020122
9808  027532  040504  040524  042440
9809  027540  051122  051117     000
9810  027545     015  041412  047514  EM6:    .ASCIZ  <15><12>/CLOCKB BR DATA ERROR/
9811  027552  045503  020102  051102
9812  027560  042040  052101  020101
9813  027566  051105  047522  000122
9814  027574  005015  046103  041517  EM7:    .ASCIZ  <15><12>/CLOCKB CR DATA ERROR/
9815  027602  041113  041440  020122
9816  027610  040504  040524  042440
9817  027616  051122  051117     000
9818  027623     015  042012  040525  EM10:   .ASCIZ  <15><12>/DUAL ADDRESS ERROR/
9819  027630  020114  042101  051104
9820  027636  051505  020123  051105
9821  027644  047522  000122
9822  027650  005015  046103  041517  EM11:   .ASCIZ  <15><12>#CLOCK A COUNT ERROR #
9823  027656  020113  020101  047503
9824  027664  047116  020124  051105
9825  027672  047522  020122     000
9826  027677     015  041412  047514  EM12:   .ASCIZ  <15><12>#CLOCK A COUNT FUNCTION ERROR #
9827  027704  045503  040440  041440
9828  027712  052517  052116  043040
9829  027720  047125  052103  047511
9830  027726  020116  051105  047522
9831  027734  020122     000
9832  027737     015  041412  047514  EM14:   .ASCIZ  <15><12>#CLOCK B COUNT FUNCTION ERROR #
9833  027744  045503  041040  041440
9834  027752  052517  052116  043040
9835  027760  047125  052103  047511
9836  027766  020116  051105  047522
9837  027774  020122     000
9838  027777     015  041412  047514  EM15:   .ASCIZ  <15><12>#CLOCK B COUNT ERROR #
9839  030004  045503  041040  041440
9840  030012  052517  052116  042440
9841  030020  051122  051117  000040
9842  030026  005015  046103  041517  EM16:   .ASCIZ  <15><12>#CLOCK A INTERRUPT ERROR #
9843  030034  020113  020101  047111
9844  030042  042524  051122  050125
9845  030050  020124  051105  047522
9846  030056  020122     000
9847  030061     015  041412  047514  EM17:   .ASCIZ  <15><12>#CLOCK B INTERRUPT ERROR #
9848  030066  045503  041040  044440
9849  030074  052116  051105  052522
9850  030102  052120  042440  051122
9851  030110  051117  000040
9852  030114  005015  046103  041517  EM20:   .ASCIZ  <15><12>#CLOCK A REPEATABILITY ERROR #
9853  030122  020113  020101  042522
9854  030130  042520  052101  041101
```

```
 9855  030136  046111  052111  020131
 9856  030144  051105  047522  020122
 9857  030152     000
 9858  030153     015  041412  047514  EM23:   .ASCIZ  <15><12>#CLOCK B REPEATABILITY ERROR #
 9859  030160  045503  041040  051040
 9860  030166  050105  040505  040524
 9861  030174  044502  044514  054524
 9862  030202  042440  051122  051117
 9863  030210  000040
 9864  030212  005015  046103  041517  EM26:   .ASCIZ  <15><12>#CLOCK ADDRESSING ERROR#
 9865  030220  020113  042101  051104
 9866  030226  051505  044523  043516
 9867  030234  042440  051122  051117
 9868  030242     000
 9869
 9870  030243     015  042412  051122  DH1:    .ASCIZ  <15><12>#ERRPC   ASR     WAS     S/B#
 9871  030250  041520  020040  040440
 9872  030256  051123  020040  020040
 9873  030264  053440  051501  020040
 9874  030272  020040  051440  041057
 9875  030300     000
 9876  030301     015  042412  051122  DH3:    .ASCIZ  <15><12>#ERRPC   ABR     WAS     S/B#
 9877  030306  041520  020040  040440
 9878  030314  051102  020040  020040
 9879  030322  053440  051501  020040
 9880  030330  020040  051440  041057
 9881  030336     000
 9882  030337     015  042412  051122  DH4:    .ASCIZ  <15><12>#ERRPC   ACR     WAS     S/B#
 9883  030344  041520  020040  040440
 9884  030352  051103  020040  020040
 9885  030360  053440  051501  020040
 9886  030366  020040  051440  041057
 9887  030374     000
 9888  030375     015  042412  051122  DH5:    .ASCIZ  <15><12>#ERRPC   BSR     WAS     S/B#
 9889  030402  041520  020040  041040
 9890  030410  051123  020040  020040
 9891  030416  053440  051501  020040
 9892  030424  020040  051440  041057
 9893  030432     000
 9894  030433     015  042412  051122  DH6:    .ASCIZ  <15><12>#ERRPC   BBR     WAS     S/B#
 9895  030440  041520  020040  041040
 9896  030446  051102  020040  020040
 9897  030454  053440  051501  020040
 9898  030462  020040  051440  041057
 9899  030470     000
 9900  030471     015  042412  051122  DH7:    .ASCIZ  <15><12>#ERRPC   BCR     WAS     S/B#
 9901  030476  041520  020040  041040
 9902  030504  051103  020040  020040
 9903  030512  053440  051501  020040
 9904  030520  020040  051440  041057
 9905  030526     000
 9906  030527     015  042412  051122  DH10:   .ASCII  <15><12>/ERROR   GOOD    BAD     GOOD    DATA READ FROM/
 9907  030534  051117  020040  043440
 9908  030542  047517  020104  020040
 9909  030550  041040  042101  020040
 9910  030556  020040  043440  047517
```

```
 9911  030564  020104  020040  042040
 9912  030572  052101  020101  042522
 9913  030600  042101  043040  047522
 9914  030606     115
 9915  030607     015  020012  050040          .ASCIZ  <15><12>/  PC    ADDR    ADDR    DATA    DUAL ADDRESS/
 9916  030614  020103  020040  040440
 9917  030622  042104  020122  020040
 9918  030630  040440  042104  020122
 9919  030636  020040  042040  052101
 9920  030644  020101  020040  042040
 9921  030652  040525  020114  042101
 9922  030660  051104  051505  000123
 9923  030666  005015  051105  050122  DH12:   .ASCIZ  <15><12>#ERRPC   ASR #
 9924  030674  020103  020040  051501
 9925  030702  020122     000
 9926  030705     015  042412  051122  DH14:   .ASCIZ  <15><12>#ERRPC   BSR#
 9927  030712  041520  020040  041040
 9928  030720  051123     000
 9929  030723     015  042412  051122  DH20:   .ASCIZ  <15><12>#ERRPC   ASR     2NDCNT  1STCNT #
 9930  030730  041520  020040  040440
 9931  030736  051123  020040  020040
 9932  030744  031040  042116  047103
 9933  030752  020124  030440  052123
 9934  030760  047103  020124     000
 9935  030765     015  042412  051122  DH23:   .ASCIZ  <15><12>#ERRPC   BSR     2NDCNT  1STCNT #
 9936  030772  041520  020040  041040
 9937  031000  051123  020040  020040
 9938  031006  031040  042116  047103
 9939  031014  020124  030440  052123
 9940  031022  047103  020124     000
 9941  031027     015  042412  051122  DH26:   .ASCIZ  <15><12>#ERRPC   CLOCK ADDR.#
 9942  031034  041520  020040  041440
 9943  031042  047514  045503  040440
 9944  031050  042104  027122     000
 9945
 9946          031056                          .EVEN
 9947
 9948  031056  001116  001324  001126  DT1:    .WORD   $ERRPC,ASR,$BDDAT,$GDDAT,0
 9949  031064  001124  000000
 9950  031070  001116  001326  001126  DT3:    .WORD   $ERRPC,ABR,$BDDAT,$GDDAT,0
 9951  031076  001124  000000
 9952  031102  001116  001330  001126  DT4:    .WORD   $ERRPC,ACR,$BDDAT,$GDDAT,0
 9953  031110  001124  000000
 9954  031114  001116  001332  001126  DT5:    .WORD   $ERRPC,BSR,$BDDAT,$GDDAT,0
 9955  031122  001124  000000
 9956  031126  001116  001334  001126  DT6:    .WORD   $ERRPC,BBR,$BDDAT,$GDDAT,0
 9957  031134  001124  000000
 9958  031140  001116  001336  001126  DT7:    .WORD   $ERRPC,BCR,$BDDAT,$GDDAT,0
 9959  031146  001124  000000
 9960  031152  001116  001120  001122  DT10:   .WORD   $ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT,0
 9961  031160  001124  001126  000000
 9962  031166  001116  001324  000000  DT12:   .WORD   $ERRPC,ASR,0
 9963  031174  001116  001332  000000  DT14:   .WORD   $ERRPC,BSR,0
 9964  031202  001116  001330  001124  DT21:   .WORD   $ERRPC,ACR,$GDDAT,$TMP0,0
 9965  031210  001162  000000
 9966  031214  001116  001330  001126  DT22:   .WORD   $ERRPC,ACR,$BDDAT,$TMP0,0
```

```
 9967  031222  001162  000000
 9968  031226  001116  001336  001124  DT24:   .WORD   $ERRPC,BCR,$GDDAT,$TMPO,0
 9969  031234  001162  000000
 9970  031240  001116  001336  001126  DT25:   .WORD   $ERRPC,BCR,$BDDAT,$TMPO,0
 9971  031246  001162  000000
 9972  031252  001116  001162  000000  DT26:   .WORD   $ERRPC,$TMPO,0
 9973
 9974  031260  000000  000000          DFO:    .WORD   0,0
 9975
 9976
 9977
 9978
 9979          000001                          .END
```

```
ABASE = 170404          875#   974   1018
ABR     001326         1044#  1319#  1420  1426  1659#  1661  1694#  1695  2368#  2369#  2371  2385#  2387
                       2413#  2414#  2416  2430#  2432  2458#  2459#  2461  2475#  2477  2503#  2504#  2506
                       2520#  2522   2548#  2549#  2551  2565#  2567  2593#  2594#  2596  2610#  2612   2638#
                       2639#  2641   2655#  2657   2683#  2684#  2686  2700#  2702   2728#  2729#  2731   2745#
                       2747   2773#  2774#  2776   2790#  2792   2818#  2819#  2821   2835#  2837   2863#  2864#
                       2866   2880#  2882   2908#  2909#  2911   2925#  2927   2953#  2954#  2956   2970#  2972
                       2998#  2999#  3001   3015#  3017   3043#  3044#  3046   3060#  3062   4081#  4118#  4161#
                       4364#  4368   4537#  4591#  4651#  4700#  4758#  4797#  4836#  4875#  4914#  4953#  4993#
                       5037#  5044#  5078#  5115#  5205#  5208#  5212   5253#  5256#  5260   5290#  5324#  5355#
                       5390#  5435#  5452   5508#  6041#  6396#  6858#  6911#  6989#  7067#  7145#  7225#  7237#
                       7330#  7342#  7435#  7447#  7540#  7552#  7645#  7657#  8802#  8964#  9950
ACDW1 = 000000          974   1020
ACDW2 = 000000          974   1021
ACPUOP= 000000          974    989
ACR     001330         1045#  1321#  1466  1472  4086  4125  4168  4543  4604  4722  4763  4802  4841
                       4880   4919   4958  4999  5047  5121  5296  5327  5358  5394  5471  5515  5756
                       6053   6866   6918  6929  6950  6996  7007  7028  7074  7085  7106  7152  7163
                       7184   7235   7247  7340  7352  7445  7457  7550  7562  7655  7667  8811  9952
                       9964   9966
ADDW0 = 000000          974   1022
ADDW1 = 000000          974   1023
ADDW10= 000000          974   1032
ADDW11= 000000          974   1033
ADDW12= 000000          974   1034
ADDW13= 000000          974   1035
ADDW14= 000000          974   1036
ADDW15= 000000          974   1037
ADDW2 = 000000          974   1024
ADDW3 = 000000          974   1025
ADDW4 = 000000          974   1026
ADDW5 = 000000          974   1027
ADDW6 = 000000          974   1028
ADDW7 = 000000          974   1029
ADDW8 = 000000          974   1030
ADDW9 = 000000          974   1031
ADEVCT= 000000          974    980
ADEVM = 000000          974   1019
AENV  = 000000          974    985
AENVM = 000000          974    986
AFATAL= 000000          974    977
AMADR1= 000000          974   1002
AMADR2= 000000          974   1006
AMADR3= 000000          974   1009
AMADR4= 000000          974   1012
AMAMS1= 000000          974    996
AMAMS2= 000000          974   1004
AMAMS3= 000000          974   1007
AMAMS4= 000000          974   1010
AMSGAD= 000000          974    982
AMSGLG= 000000          974    983
AMSGTY= 000000          974    976
AMTYP1= 000000          974    997
AMTYP2= 000000          974   1005
AMTYP3= 000000          974   1008
AMTYP4= 000000          974   1011
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
APASS = 000000          974     979
APRIOR= 000006          877#    974     1015
APRITY  001350         1057#   1309*    1337    6243    6299    8635*
APTCSU= 000040         9603    9709#
APTENV= 000001         9258    9596     9665    9707#
APTSIZ= 000200         1295    9706#
APTSPO= 000100         9598    9667     9708#
ASR     001324         1043#   1307*    1317    1374    1380    1734*   1735    1748*   1828*   1829*   1831    1845*   1847
                       1873*   1874*    1876    1890*   1892    1918*   1919*   1921    1935*   1937    1963*   1964*   1966
                       1980*   1982     2008*   2009*   2011    2025*   2027    2053*   2054*   2056    2070*   2072    2098*
                       2099*   2101     2115*   2117    2143*   2144*   2146    2160*   2162    2188*   2189*   2191    2205*
                       2207    2233*    2234*   2236    2250*   2252    2278*   2279*   2281    2295*   2297    2323*   2324*
                       2326    2340*    2342    3876*   3877*   3882    3925*   3927*   3931*   3937*   3939    4080*   4117*
                       4160*   4328*    4332    4468*   4469*   4471    4500*   4501*   4502    4517*   4536*   4538*   4540*
                       4590*   4596*    4598*   4650*   4652*   4653*   4655    4669    4699*   4701*   4702*   4708    4757*
                       4759*   4766     4796*   4798*   4805    4835*   4837*   4844    4874*   4876*   4883    4913*   4915*
                       4922    4952*    4954*   4961    4992*   4994*   5002    5036*   5040*   5077*   5079*   5080*   5083
                       5114*   5116*    5117*   5162*   5163*   5164*   5168    5204*   5207*   5209*   5252*   5255*   5257*
                       5289*   5323*    5325*   5326*   5354*   5356*   5357*   5389*   5391*   5434*   5437*   5438*   5506*
                       5509*   5517     5529*   5559*   5609*   5620*   5669*   5675*   5719*   5746*   5749*   5787*   5829*
                       5871*   5913*    5955*   5997*   6039*   6047*   6048*   6154*   6165*   6254*   6309*   6334*   6352*
                       6354*   6355*    6394*   6397*   6398*   6417*   6434*   6437*   6438*   6473*   6548*   6745*   6857*
                       6859*   6910*    6913*   6917*   6946*   6988*   6991*   6995*   7024*   7066*   7069*   7073*   7102*
                       7144*   7147*    7151*   7180*   7223*   7226*   7229*   7234*   7238*   7241*   7246*   7328*   7331*
                       7334*   7339*    7343*   7346*   7351*   7433*   7436*   7439*   7444*   7448*   7451*   7456*   7538*
                       7541*   7544*    7549*   7553*   7556*   7561*   7643*   7646*   7649*   7654*   7658*   7661*   7666*
                       7753*   7760*    7792*   7835*   7842*   7874*   7917*   7924*   7956*   7999*   8006*   8038*   8081*
                       8184*   8287*    8390*   8493*   8588*   8592    8633*   8717*   8721*   8727    8799*   8803*   8804*
                       8882*   8885*    8895    8961*   8966*   8968    8977    9044*   9060    9948    9962
ASWREG= 000000          974     987
ATESTN= 000000          974     978
AUNIT = 000000          974     981
AUSWR = 000000          974     988
AVECP2  001342         1052#   1331*    6158*   6436*   6474    6475*
AVECT   001340         1051#   1308*    1329    6157*   6187*   6242*   6298*   6353*   6395*   6435*   6474*   8613*   8634*
AVECT1= 000344          876#    974     1013
AVECT2= 000000          974     1014
BBR     001334         1048#   1325*    1558    1564    1797*   1798    3503*   3504*   3506    3520*   3522    3548*   3549*
                       3551    3565*    3567    3593*   3594*   3596    3610*   3612    3638*   3639*   3641    3655*   3657
                       3683*   3684*    3686    3700*   3702    3728*   3729*   3731    3745*   3747    3773*   3774*   3776
                       3790*   3792     3818*   3819*   3821    3835*   3837    4205*   4243*   4286*   4435*   4439    5556*
                       5611*   5671*    5721*   5748*   5789*   5831*   5873*   5915*   5957*   5999*   6042*   6747*   7754*
                       7836*   7918*    8000*   8083*   8095*   8186*   8198*   8289*   8301*   8392*   8404*   8495*   8507*
                       9047*   9956
BCR     001336         1049#   1327*    1604    1610    4210    4250    4293    5560    5626    5690    5726    5796    5838
                       5880    5922     5964    6006    7761    7773    7796    7843    7855    7878    7925    7937    7960
                       8007    8019     8042    8093    8105    8196    8208    8299    8311    8402    8414    8505    8517
                       9958    9968     9970
BIT0  = 000001          859#    2324     2325    2340    2369    2370    2385    3459    3460    3475    3504    3505    3520
                       5558    6397     6750
BIT00 = 000001          849#    859      4502    4708    5083
BIT01 = 000002          848#    858
BIT02 = 000004          847#    857
BIT03 = 000010          846#    856
BIT04 = 000020          845#    855
BIT05 = 000040          844#    854      4655    4766    4805    4844    4883    4922    4961    5002    8968
```

DZKWK.CMB CROSS REFEPENCE TABLE -- USER SYMBOLS

```
BIT06 = 000100   843#  853  6254
BIT07 = 000200   842#  852  4669  5168  9051
BIT08 = 000400   841#  851  9422
BIT09 = 001000   840#  850  9268  9430
BIT1  = 000002   858#  2279  2280  2295  2414  2415  2430  3413  3414  3429  3549  3550  3565
                 5558  6750
BIT10 = 002000   839#  2819  2820  2835  5164  5209  5257  5326  5357  5391  6166  6255  6310
                 6438  8721  8885  9246
BIT11 = 004000   838#  2864  2865  2880  3091  3092  3107  5507  5555  5559  5610  5620  5675
                 6048  6912  6917  6946  6990  6995  7024  7068  7073  7102  7146  7151  7180
                 7755  7760  7792  7837  7842  7874  7919  7924  7956  8001  8005  8038  9437
BIT12 = 010000   837#  2909  2910  2925  4469  4501  4540  4598  4653  4702  5080  5117  5438
                 6355  6398  6859  8804
BIT13 = 020000   836#  1919  1920  1935  2954  2955  2970  4500  6857  8744  8826  8909  8992
                 9075  9253
BIT14 = 040000   835#  1874  1875  1890  2999  3000  3015  4624  5646  6354  9408
BIT15 = 100000   834#  1829  1830  1845  3044  3045  3060  8727
BIT2  = 000004   857#  2234  2235  2250  2459  2460  2475  3367  3368  3383  3594  3595  3610
                 6397
BIT3  = 000010   856#  2189  2190  2205  2504  2505  2520  3321  3322  3337  3639  3640  3655
                 6397
BIT4  = 000020   855#  2549  2550  2565  3275  3276  3291  3684  3685  3700
BIT5  = 000040   854#  2144  2145  2160  2594  2595  2610  3229  3230  3245  3729  3730  3745
                 5558  7755  7837  7919  8001
BIT6  = 000100   853#  2099  2100  2115  2639  2640  2655  3183  3184  3199  3774  3775  3790
                 6165  6309  6397  6558  6647  6704  6750
BIT7  = 000200   852#  2054  2055  2070  2684  2685  2700  3137  3138  3153  3819  3820  3835
BIT8  = 000400   851#  2009  2010  2025  2729  2730  2745  8803
BIT9  = 001000   850#  1964  1965  1980  2774  2775  2790  6559  6648  6705  8803
BPRITY  001352  1058#  1337*  6636  6694
BPTVEC= 000014   866#
BSR     001332  1047#  1323*  1512  1518  1765*  1766  3090*  3091*  3093  3107*  3109  3136*  3137*
                 3139  3153*  3155  3182*  3183*  3185  3199*  3201  3228*  3229*  3231  3245*  3247
                 3274*  3275*  3277  3291*  3293  3320*  3321*  3323  3337*  3339  3366*  3367*  3369
                 3383*  3385  3412*  3413*  3415  3429*  3431  3458*  3459*  3461  3475*  3477  3981*
                 3982*  3987  4030*  4032*  4036*  4042*  4044  4204*  4242*  4285*  4399*  4403  5035
                 5161  5203  5251  5433*  5436*  5489*  5507*  5555*  5558*  5610*  5616*  5618*  5670*
                 5672*  5674*  5677  5720*  5747*  5788*  5790*  5799  5830*  5832*  5841  5872*  5874*
                 5883  5914*  5916*  5925  5956*  5958*  5967  5998*  6000*  6009  6040*  6044*  6046*
                 6155*  6166*  6255*  6310*  6351*  6393*  6433*  6549*  6558*  6559*  6647*  6648*  6672*
                 6704*  6705*  6746*  6750*  6781*  6909*  6912*  6987*  6990*  7065*  7068*  7143*  7146*
                 7224*  7329*  7434*  7539*  7644*  7752*  7755*  7756*  7757*  7834*  7837*  7838*  7839*
                 7916*  7919*  7920*  7921*  7998*  8001*  8002*  8003*  8082*  8084*  8087*  8092*  8096*
                 8099*  8104*  8163*  8185*  8187*  8190*  8195*  8199*  8202*  8207*  8266*  8288*  8290*
                 8293*  8298*  8302*  8305*  8310*  8369*  8391*  8393*  8396*  8401*  8405*  8408*  8413*
                 8472*  8494*  8496*  8499*  8504*  8508*  8511*  8516*  8575*  8718*  8800*  8883*  8962*
                 9045*  9049*  9051  9954  9963
BVECT   001344  1054#  1333*  6551*  6581*  6635*  6693*  6748*  6783*  6826*
BVECT2  001346  1055#  1335*  6552*  6749*  6826  6827*
CKSWR = 104405  9785#
CR    = 000015   774#  9642  9652
CRLF  = 000200   775#  9613  9652
DDISP = 177570   781#  950  1282
DFO     031260  1083  1091  1099  1107  1115  1123  1131  1140  1148  1156  1161  1162  1163
                 1164  1171  1179  1187  1195  1202  1210  1218  1226  1234  1242  1250  9974#
DH1     030243  1081  1089  9870#
```

DZKWK.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

```
DH10    030527          1137    9906#
DH12    030666          1154    1185    9923#
DH14    030705          1169    1193    9926#
DH20    030723          1200    9929#
DH23    030765          1224    9935#
DH26    031027          1248    9941#
DH3     030301          1097    9876#
DH4     030337          1105    1146    1208    1216    9882#
DH5     030375          1113    9888#
DH6     030433          1121    9894#
DH7     030471          1129    1177    1232    1240    9900#
DISPLA  001140           950#   1282*   1288*   9245*   9452*
DISPRE  000174           751#   1288
DSWR  = 177570           780#    949    1281
DT1     031056          1082    1090    1201    1225    9948#
DT10    031152          1139    9960#
DT12    031166          1155    1186    9962#
DT14    031174          1170    1194    9963#
DT21    031202          1209    9964#
DT22    031214          1217    9966#
DT24    031226          1233    9968#
DT25    031240          1241    9970#
DT26    031252          1249    9972#
DT3     031070          1098    9950#
DT4     031102          1106    1147    9952#
DT5     031114          1114    9954#
DT6     031126          1122    9956#
DT7     031140          1130    1178    9958#
EMTVEC= 000030           869#   1266*   1267*
EM1     027356          1080    9789#
EM10    027623          1136    9818#
EM11    027650          1145    1207    1215    9822#
EM12    027677          1153    9826#
EM14    027737          1168    9832#
EM15    027777          1176    1231    1239    9838#
EM16    030026          1184    9842#
EM17    030061          1192    9847#
EM2     027411          1088    9794#
EM20    030114          1199    9852#
EM23    030153          1223    9858#
EM26    030212          1247    9864#
EM3     027440          1096    9798#
EM4     027467          1104    9802#
EM5     027516          1112    9806#
EM6     027545          1120    9810#
EM7     027574          1128    9814#
ERRVEC= 000004           862#   1279    1280*   1291*   1370    1371*   1397*   1416    1417*   1443*   1462    1463*   1489*
                        1508    1509*   1535*   1554    1555*   1581*   1600    1601*   1627*   8589    8590*   8612*   8618*
                        9413    9414*   9416*   9419*
GNS   = ****** U         750    1304    6143    6238    6537    6631    8597    8603    8608    8624    8630    8753    8760
                        8835    8842    8918    8925    9001    9008    9084    9091    9106    9114    9780    9781    9782
                        9783    9784    9785    9786    9787    9788
HT    = 000011           772#   9611    9652
IOTRD   024270          9101#   9788
IOTT  = 104410          6475    6827    9788#
IOTVEC= 000020           867#   1264*   1265*
```

```
LF     = 000012          773#   9646    9652
LOOP     002174         1313#   8614    8673
LS210    023446          757    8709#
LS214    023560          758    8791#
LS220    023704          759    8874#
LS224    024014          760    8953#
LS230    024142          761    9036#
P      = 000012         2355#   2400#   2445#   2490#   2535#   2580#   2625#   2670#   2715#   2760#   2805#   2850#   2895#
                        2940#   2985#   3030#   3075#   3490#   3535#   3580#   3625#   3670#   3715#   3760#   3805#   3850#
                        5228#   5276#
PC     =%000007          793#   8657*   8660*   8667*   8672    9255*   9261*   9314*   9505*   9601*   9620*   9627*   9634*
                        9648*   9650*   9684*   9701*
PIRQ   = 177772          779#
PIRQVE= 000240           873#
PR0    = 000000          796#
PR1    = 000040          797#
PR2    = 000100          798#
PR3    = 000140          799#
PR4    = 000200          800#
PR5    = 000240          801#
PR6    = 000300          802#
PR7    = 000340          803#
PS     = 177776          776#    777
PSW    = 177776          777#
PWRVEC= 000024           868#   1270*   1271*   9715*   9716*   9725*   9731*   9743*   9744*
RDCHR  = 104406         9480    9546    9786#
RDLIN  = 104407         9787#
RESVEC= 000010           863#
RSTART   002320          756    1338#
R0     =%000000          784#   1314*   1315*   1317*   1318*   1319    1320*   1321    1322*   1323    1324*   1325    1326*
                        1327    1329*   1330*   1331    1332*   1333    1334*   1335    4760*   4761*   4799*   4800*   4838*
                        4839*   4877*   4878*   4916*   4917*   4955*   4956*   4996*   4997*   5510*   5511*   5751*   5752*
                        5792*   5793*   5834*   5835*   5876*   5877*   5918*   5919*   5960*   5961*   6002*   6003*   6243*
                        6244*   6245*   6246*   6247*   6248*   6249*   6250    6299*   6300*   6301*   6302*   6303*   6304*
                        6305    6636*   6637*   6638*   6639*   6640*   6641*   6642*   6643    6694*   6695*   6696*   6697*
                        6698*   6699*   6700    6915*   6924*   6944*   6947*   6993*   7002*   7022*   7025*   7071*   7080*
                        7100*   7103*   7149*   7158*   7178*   7181*   7227*   7231*   7239*   7243*   7249*   7250*   7254*
                        7257    7332*   7336*   7344*   7348*   7354*   7355*   7359*   7362    7437*   7441*   7449*   7453*
                        7459*   7460*   7464*   7467    7542*   7546*   7554*   7558*   7564*   7565*   7569*   7572    7647*
                        7651*   7659*   7663*   7669*   7670*   7674*   7677    7758*   7768*   7790*   7793*   7840*   7850*
                        7872*   7875*   7922*   7932*   7954*   7957*   8004*   8014*   8036*   8039*   8085*   8089*   8097*
                        8101*   8107*   8108*   8112*   8115    8188*   8192*   8200*   8204*   8210*   8211*   8215*   8218
                        8291*   8295*   8303*   8307*   8313*   8314*   8318*   8321    8394*   8398*   8406*   8410*   8416*
                        8417*   8421*   8424    8497*   8501*   8509*   8513*   8519*   8520*   8524*   8527    8664*   8667
                        9286    9287*   9288*   9295*   9296*   9297*   9298*   9299*   9300    9305    9310*   9312*   9316
                        9318    9338    9348*   9352    9368    9369    9382*   9594    9595*   9600    9605    9608*   9661
                        9669*   9673    9674    9676*   9677*   9678    9700*   9717    9742*   9763    9764*   9765    9766*
                        9767*   9768*   9769*
R1     =%000001          785#   9339    9352*   9353    9357    9381*   9662    9699*   9718    9741*
R2     =%000002          786#   9340    9351*   9355*   9358    9365*   9366*   9367    9372*   9380*   9719    9740*
R3     =%000003          787#   9183    9192*   9198*   9199*   9202*   9207*   9208*   9209    9218*   9341    9349*   9350*
                        9364*   9367*   9376*   9377*   9379*   9542    9543*   9544    9547*   9548    9552    9554    9556*
                        9558*   9720    9739*
R4     =%000004          788#   9184    9186*   9187*   9188*   9189    9190*   9204    9206*   9214*   9217*   9721    9738*
R5     =%000005          789#   9185    9191*   9193*   9195*   9196*   9197*   9198    9216*   9342    9344*   9346*   9353*
                        9357*   9372    9378*   9722    9737*
```

```
R6      =%000006        790*  792   1258*  1259*  1260   1379*  1425*  1471*  1517*  1563*  1609*  6186*  6197*
                        6275* 6318* 6375*  6416*  6471*  6580*  6591*  6669*  6713*  6799*  8617*  9102   9117*
                        9119
R7      =%000007        791*  793
SP      =%000006        792*  1262* 1279*  1289*  1291   1338*  1339*  1340*  1370*  1397   1416*  1443   1462*
                        1489  1508* 1535   1554*  1581   1600*  1627   6160*  6161*  6189*  6190*  6250*  6251*
                        6305* 6306* 6358*  6359*  6402*  6403*  6442*  6443*  6553*  6554*  6583*  6584*  6643*
                        6644* 6700* 6701*  6751*  6752*  6779*  6790*  6791*  8599*  8610*  8626*  9109*  9121*
                        9175* 9176  9177   9178*  9183*  9184*  9185*  9191   9216   9217   9218   9219*  9220*
                        9250  9270* 9273*  9286*  9291*  9312   9316*  9338*  9339*  9340*  9341*  9342*  9343*
                        9344  9347* 9360   9362*  9364   9374   9376   9378   9379   9380   9381   9382   9384*
                        9385* 9413* 9416   9418   9419   9448   9449   9453*  9468*  9469*  9470   9475*  9478*
                        9479* 9482  9485*  9488   9490   9492   9493*  9495   9497   9499   9500*  9501*  9502*
                        9506* 9507  9509*  9510*  9511*  9512*  9513*  9523*  9524*  9527*  9528*  9529   9531
                        9533* 9542* 9547   9558   9559*  9560*  9561*  9594*  9595   9605*  9607   9608   9609*
                        9611  9613  9615   9621   9623*  9625*  9633*  9637   9641   9642   9646   9661*  9662*
                        9669  9670* 9681   9682*  9683*  9693   9694*  9699   9700   9717*  9718*  9719*  9720*
                        9721* 9722* 9723*  9724   9732*  9736   9737   9738   9739   9740   9741   9742   9763*
                        9764
STACK = 001100          767*  1262  1338   6779
START   001634          754   1256*
STKLMT= 177774          778*
SWR     001136          949*  1281* 1283   1287*  1297*  4624   5646   8744   8826   8909   8992   9075   9246
                        9253  9265  9268   9408   9422   9424   9430   9437   9464   9492*  9499*  9723   9736*
SWREG   000176          752*  1287  9464   9475
SW0   = 000001          831*
SW00  = 000001          821*  831
SW01  = 000002          820*  830
SW02  = 000004          819*  829
SW03  = 000010          818*  828
SW04  = 000020          817*  827
SW05  = 000040          816*  826
SW06  = 000100          815*  825
SW07  = 000200          814*  824
SW08  = 000400          813*  823
SW09  = 001000          812*  822
SW1   = 000002          830*
SW10  = 002000          811*
SW11  = 004000          810*
SW12  = 010000          809*
SW13  = 020000          808*
SW14  = 040000          807*
SW15  = 100000          806*
SW2   = 000004          829*
SW3   = 000010          828*
SW4   = 000020          827*
SW5   = 000040          826*
SW6   = 000100          825*
SW7   = 000200          824*
SW8   = 000400          823*
SW9   = 001000          822*
TBITVE= 000014          864*
TE      023036          8584*
TKVEC = 000060          871*
TPVEC = 000064          872*
TRAPVE= 000034          870*  1268*  1269*
```

```
TRTVEC= 000014            865#
TST1    002336           1362#
TST10   003012           1690#
TST100  011216           4201#
TST101  011254           4239#
TST102  011324           4282#
TST103  011374           4324#
TST104  011432           4360#
TST105  011470           4395#
TST106  011526           4431#
TST107  011564           4466#
TST11   003044           1730#
TST110  011610           4498#
TST111  011644           4534#
TST112  011720           4583#
TST113  012040           4648#
TST114  012116           4697#
TST115  012174           4754#
TST116  012250           4793#
TST117  012324           4832#
TST12   003106           1761#
TST120  012400           4871#
TST121  012454           4910#
TST122  012530           4949#
TST123  012604           4989#
TST124  012656           5033#
TST125  012716           5075#
TST126  012760           5112#
TST127  013020           5159#
TST13   003144           1793#
TST130  013060           5201#
TST131  013146           5249#
TST132  013234           5286#
TST133  013276           5321#
TST134  013336           5352#
TST135  013376           5387#
TST136  013432           5431#
TST137  013550           5503#
TST14   003202           1826#
TST140  013634           5553#
TST141  013674           5601#
TST142  014026           5667#
TST143  014112           5716#
TST144  014160           5743#
TST145  014226           5784#
TST146  014304           5826#
TST147  014362           5868#
TST15   003300           1871#
TST150  014440           5910#
TST151  014516           5952#
TST152  014574           5994#
TST153  014652           6037#
TST154  014734           6129#
TST155  015150           6227#
TST156  015320           6295#
TST157  015422           6349#
```

TST16	003376	1916#
TST160	015502	6391#
TST161	015574	6431#
TST162	015702	6523#
TST163	016116	6620#
TST164	016272	6690#
TST165	016370	6743#
TST166	016532	6855#
TST167	016602	6906#
TST17	003474	1961#
TST170	016754	6984#
TST171	017126	7062#
TST172	017300	7140#
TST173	017452	7220#
TST174	017654	7325#
TST175	020056	7430#
TST176	020260	7535#
TST177	020462	7640#
TST2	002432	1412#
TST20	003572	2006#
TST200	020664	7749#
TST201	021042	7831#
TST202	021220	7913#
TST203	021376	7995#
TST204	021554	8078#
TST205	021766	8181#
TST206	022200	8284#
TST207	022412	8387#
TST21	003670	2051#
TST210	022624	8490#
TST22	003766	2096#
TST23	004064	2141#
TST24	004162	2186#
TST25	004260	2231#
TST26	004356	2276#
TST27	004454	2321#
TST3	002504	1458#
TST30	004552	2366#
TST31	004650	2411#
TST32	004746	2456#
TST33	005044	2501#
TST34	005142	2546#
TST35	005240	2591#
TST36	005336	2636#
TST37	005434	2681#
TST4	002556	1504#
TST40	005532	2726#
TST41	005630	2771#
TST42	005726	2816#
TST43	006024	2861#
TST44	006122	2906#
TST45	006220	2951#
TST46	006316	2996#
TST47	006414	3041#
TST5	002630	1550#
TST50	006512	3088#

```
TST51   006610          3134#
TST52   006706          3180#
TST53   007004          3226#
TST54   007102          3272#
TST55   007200          3318#
TST56   007276          3364#
TST57   007374          3410#
TST6    002702          1596#
TST60   007472          3456#
TST61   007570          3501#
TST62   007666          3546#
TST63   007764          3591#
TST64   010062          3636#
TST65   010160          3681#
TST66   010256          3726#
TST67   010354          3771#
TST7    002754          1656#
TST70   010452          3816#
TST71   010550          3873#
TST72   010622          3922#
TST73   010704          3978#
TST74   010756          4028#
TST75   011040          4077#
TST76   011076          4114#
TST77   011146          4157#
TYPDS = 104404          9784#
TYPE  = 104400          1302    6141    6236    6535    6629    8595    8601    8606    8622    8628    8663    8751    8758
                        8833    8840    8916    8923    8999    9006    9082    9089    9104    9112    9210    9248    9256
                        9285    9302    9304    9307    9309    9313    9320    9383    9472    9474    9477    9484    9503
                        9550    9553    9557    9616    9745    9780#
TYPOC = 104401          8600    8611    8627    9110    9122    9293    9317    9476    9781#
TYPON = 104403          9783#
TYPOS = 104402          9782#
$APTHD  001000           909     915#
$ARGI   002056          1293#
$ASTAT= ****** U        9687    9702
$ATYC   026712          9658    9660#
$ATYI   026666          9656#
$ATY3   026674          9601    9657#
$ATY4   026704          9261    9659#
$BASE   001254          1018#   1307    8633
$BOAOR  001122           943#   9960
$BOOAT  001126           945#   1260    1369#   1415#   1461#   1507#   1553#   1599#   1661#   1695#   1735#   1766#   1798#
                        1831#   1832    1847#   1876#   1877    1892#   1921#   1922    1937#   1966#   1967    1982#   2011#
                        2012    2027#   2056#   2057    2072#   2101#   2102    2117#   2146#   2147    2162#   2191#   2192
                        2207#   2236#   2237    2252#   2281#   2282    2297#   2326#   2327    2342#   2371#   2372    2387#
                        2416#   2417    2432#   2461#   2462    2477#   2506#   2507    2522#   2551#   2552    2567#   2596#
                        2597    2612#   2641#   2642    2657#   2686#   2687    2702#   2731#   2732    2747#   2776#   2777
                        2792#   2821#   2822    2837#   2866#   2867    2882#   2911#   2912    2927#   2956#   2957    2972#
                        3001#   3002    3017#   3046#   3047    3062#   3093#   3094    3109#   3139#   3140    3155#   3185#
                        3186    3201#   3231#   3232    3247#   3277#   3278    3293#   3323#   3324    3339#   3369#   3370
                        3385#   3415#   3416    3431#   3461#   3462    3477#   3506#   3507    3522#   3551#   3552    3567#
                        3596#   3597    3612#   3641#   3642    3657#   3686#   3687    3702#   3731#   3732    3747#   3776#
                        3777    3792#   3821#   3822    3837#   3882#   3884    3887    3939#   3940    3942    3987#   3989
                        3992    4044#   4045    4047    4086#   4125#   4127    4168#   4170    4210#   4250#   4252    4293#
                        4295    4332#   4368#   4403#   4439#   4543#   4544    4604#   4606    5212#   5213    5260#   5261
```

```
                        5296*  5452*  5453   5471*  5472   5626*  5628   5726*  6866*  6867   6929*  6931   6950*
                        6951   7007*  7009   7028*  7029   7085*  7087   7106*  7107   7163*  7165   7184*  7185
                        7247*  7250   7292   7352*  7355   7397   7457*  7460   7502   7562*  7565   7607   7667*
                        7670   7712   7773*  7775   7796*  7797   7855*  7857   7878*  7879   7937*  7939   7960*
                        7961   8019*  8021   8042*  8043   8105*  8108   8149   8208*  8211   8252   8311*  8314
                        8355   8414*  8417   8458   8517*  8520   8561   9948   9950   9552   9954   9956   9958
                        9960   9966   9970
$BELL   001170           965#   9248   9276
$CDW1   001260          1020#
$CDW2   001262          1021#
$CHARC  026662          9618*  9628*  9635   9644*  9649#
$CKSWR  025712          9464#  9785
$CMTAG  001100           931#  1257   1258   1266   1272   1273
$CM1  = 000001           961#   962#
$CM2  = 000002           961#   962#
$CM3  = 000001           959#   961
$CM4  = 000001           962#   963#
$CNTLG  026355          9472   9567#
$CNTLU  026350          9484   9566#
$CPUOP  001226           989#
$CRLF   001175           967#  9256   9276   9285   9304   9309   9313   9503   9566   9617   9652
$DBLK   025424          9349   9383   9391#
$DDW0   001264          1022#
$DDW1   001266          1023#
$DDW10  001310          1032#
$DDW11  001312          1033#
$DDW12  001314          1034#
$DDW13  001316          1035#
$DDW14  001320          1036#
$DDW15  001322          1037#
$DDW2   001270          1024#
$DDW3   001272          1025#
$DDW4   001274          1026#
$DDW5   001276          1027#
$DDW6   001300          1028#
$DDW7   001302          1029#
$DDW8   001304          1030#
$DDW9   001306          1031#
$DEVCT  001210           980#  1310*  8599   8605*  8610   8619   8626   8636*
$DEVM   001256          1019#
$DOAGN  023424          8659   8665   8671#
$DTBL   025414          9352   9387#
$ENDAD  023414           894   8667#
$ENDCT  023374          8661#
$ENDMG  023433          8663   8675#
$ENULL  023430          8674#
$ENV    001220           985#  9258   9596   9665   9689
$ENVM   001221           986#  1295   8585   9598   9603   9667
$EOP    023340          8651#
$EOPCT  023366          8658#  8662
$ERFLG  001103           934#  9243*  9276   9398   9426   9428   9434*  9456
$ERMAX  001115           940#  1274*  9428   9451*  9456
$ERROR  024672          1266   9242#
$ERRPC  001116           941#  9250*  9251*  9252   9276   9291   9948   9950   9952   9954   9956   9958   9960
                        9962   9963   9964   9966   9968   9970   9972
$ERRTB  001354          1076#  9299
```

```
$ERRTY  025054          9255    9284#
$ERTTL  001112           938#   9249*   9276
$ESCAP  001166           964#   1273*   9271    9273    9276    9450*
$ETABL  001220           984#
$ETEND  001324           921    1040#
$FATAL  001202           977#   9693*
$FFLG   027132          9656*   9659*   9687    9696*   9704#
$FILLC  001154           957#   9621    9652
$FILLS  001153           956#   9652
$GDAOR  001120           942#   9960
$GDDAT  001124           944#   1368*   1414*   1460*   1506*   1552*   1598*   1660*   1693*   1733*   1734    1764*   1765
                        1796*   1797    1830*   1832    1846*   1875*   1877    1891*   1920*   1922    1936*   1965*   1967
                        1981*   2010*   2012    2026*   2055*   2057    2071*   2100*   2102    2116*   2145*   2147    2161*
                        2190*   2192    2206*   2235*   2237    2251*   2280*   2282    2296*   2325*   2327    2341*   2370*
                        2372    2386*   2415*   2417    2431*   2460*   2462    2476*   2505*   2507    2521*   2550*   2552
                        2566*   2595*   2597    2611*   2640*   2642    2656*   2685*   2687    2701*   2730*   2732    2746*
                        2775*   2777    2791*   2820*   2822    2836*   2865*   2867    2881*   2910*   2912    2926*   2955*
                        2957    2971*   3000*   3002    3016*   3045*   3047    3061*   3092*   3094    3108*   3138*   3140
                        3154*   3184*   3186    3200*   3230*   3232    3246*   3276*   3278    3292*   3322*   3324    3338*
                        3368*   3370    3384*   3414*   3416    3430*   3460*   3462    3476*   3505*   3507    3521*   3550*
                        3552    3566*   3595*   3597    3611*   3640*   3642    3656*   3685*   3687    3701*   3730*   3732
                        3746*   3775*   3777    3791*   3820*   3822    3836*   3884*   3885*   3940*   3941*   3989*   3990*
                        4045*   4046*   4085*   4123*   4127    4166*   4170    4209*   4248*   4252    4291*   4295    4327*
                        4363*   4398*   4434*   4542*   4544    4586*   4591    4601*   4606    4621    4626*   5210*   5213
                        5258*   5261    5292*   5451*   5604*   5611    5623*   5628    5643    5648*   5725*   6865*   6867
                        6927*   6931    6951    7005*   7009    7029    7083*   7087    7107    7161*   7165    7185    7235*
                        7249    7275    7340*   7354    7380    7445*   7459    7485    7550*   7564    7590    7655*   7669
                        7695    7771*   7775    7797    7853*   7857    7879    7935*   7939    7961    8017*   8021    8043
                        8093*   8107    8132    8196*   8210    8235    8299*   8313    8338    8402*   8416    8441    8505*
                        8519    8544    9948    9950    9952    9954    9956    9958    9960    9964    9968
$GET42  023404          8664#
$HD   = 000001           730     731
$HIBTS  001000           916#
$ICNT   001104           935#   8711*   8747*   8793*   8829*   8876*   8912*   8955*   8995*   9038*   9078*   9441*   9442
                        9444*   9455
$ILLUP  027274          9715    9731    9748#
$ITEMB  001114           939#   9252*   9260    9276    9288
$LF     001176           968#   9276    9557    9566    9652
$LFLG   027131          9697*   9703#
$LPAOR  001106           936#   1275*   1364*   6139*   6235*   6533*   6628*   9432*   9448*   9453    9455
$LPERR  001110           937#   1276*   6137*   6234*   6531*   6627*   9270    9432    9449*   9455
$MAOR1  001232          1002#
$MAOR2  001236          1006#
$MAOR3  001242          1009#
$MAOR4  001246          1012#
$MAIL   001200           917     921     975#    1293    9258    9447    9596
$MANS1  001230           996#
$MANS2  001234          1004#
$MANS3  001240          1007#
$MANS4  001244          1010#
$MBAOR  001002           917#
$MFLG   027130          9657*   9663    9698*   9702#
$MNEW   026373          9477    9570#
$MSGAD  001214           982#   9673*   9676
$MSGLG  001216           983#   9678*
$MSGTY  001200           976#   9671    9679*   9691    9695*
```

```
$MSWR   026362          9474    9568#
$MTYP1  001231           997#
$MTYP2  001235          1005#
$MTYP3  001241          1008#
$MTYP4  001245          1011#
$MXCNT  025710          9445    9455#
$NULL   001152           955#   9623    9652
$NWTST= 000001          1350#   1352    1400#   1402    1446#   1448    1492#   1494    1538#   1540    1584#   1586    1631#
                        1633    1677#   1679    1711#   1713    1751#   1753    1784#   1786    1816#   1818    1861#   1863
                        1906#   1908    1951#   1953    1996#   1998    2041#   2043    2086#   2088    2131#   2133    2176#
                        2178    2221#   2223    2266#   2268    2311#   2313    2356#   2358    2401#   2403    2446#   2448
                        2491#   2493    2536#   2538    2581#   2583    2626#   2628    2671#   2673    2716#   2718    2761#
                        2763    2806#   2808    2851#   2853    2896#   2898    2941#   2943    2986#   2988    3031#   3033
                        3078#   3080    3124#   3126    3170#   3172    3216#   3218    3262#   3264    3308#   3310    3354#
                        3356    3400#   3402    3446#   3448    3491#   3493    3536#   3538    3581#   3583    3626#   3628
                        3671#   3673    3716#   3718    3761#   3763    3806#   3808    3856#   3858    3905#   3907    3951#
                        3963    4011#   4013    4065#   4067    4102#   4104    4145#   4147    4189#   4191    4227#   4229
                        4270#   4272    4314#   4316    4349#   4351    4385#   4387    4420#   4422    4457#   4459    4488#
                        4490    4520#   4522    4564#   4566    4636#   4638    4685#   4687    4744#   4746    4783#   4785
                        4822#   4824    4861#   4863    4900#   4902    4939#   4941    4978#   4980    5021#   5023    5064#
                        5066    5098#   5100    5145#   5147    5182#   5184    5230#   5232    5279#   5281    5311#   5313
                        5342#   5344    5374#   5376    5409#   5411    5491#   5493    5535#   5537    5585#   5587    5654#
                        5656    5706#   5708    5740#   5774#   5776    5816#   5818    5858#   5860    5900#   5902    5942#
                        5944    5984#   5986    6026#   6028    6084#   6086    6212#   6214    6280#   6282    6337#   6339
                        6381#   6383    6421#   6423    6478#   6480    6606#   6608    6675#   6677    6731#   6733    6836#
                        6838    6890#   6892    6968#   6970    7046#   7048    7124#   7126    7207#   7209    7312#   7314
                        7417#   7419    7522#   7524    7627#   7629    7732#   7734    7814#   7816    7896#   7898    7978#
                        7980    8065#   8067    8168#   8170    8271#   8273    8374#   8376    8477#   8479
$OCMT   024666          9182#   9211#   9224#
$OMOLE  024670          9177#   9181#   9186    9189#   9200#   9226#
$OVER   025674          9409    9425    9433    9443    9452#
$PASS   001206           979#   1294#   1311#   6133    6230    6527    6623    8655#   8656#   8674    8712#   8756#   8794#
                        8838#   8877#   8921#   8956#   9004#   9039#   9087#   9439    9456
$PASTM  001006           919#
$POWER  027302          9746    9751#
$PRIOR  001252          1015#   1309    8635
$PWRDN  027134          1270    9715#   9743
$PWRMG  027270          9746#
$PWRUP  027206          9725    9731#
$QUES   001174           966#   9276    9550    9566    9652
$RDCHR  026150          9523#   9786
$RDDEC= ****** U        9788
$RDLIN  026232          9542#   9787
$RDOCT= ****** U        9788
$RDSZ = 000010          9535#
$REGAD  001156           959#
$REGD   001160           961#
$RTNAD  023426          8673#
$R2A  = ****** U        9788
$SAVRE= ****** U        9788
$SAVR6  027300          9724#   9732    9733#   9734#   9750#
$SCOPE  025434          1264    9407#
$SETUP= 000017          1255#   1263    1264    1266    1268    1270    1272    1273    1275    1292    8653    9243    9268
                        9275    9408
$STUP = 177777          1255#
$SVLAD  025640          9417    9446#
```

$SVPC = 000234 892# 897
$SWR = 167400 719# 730 736 737 738 739 740 741 742 963 964 965 1272
1273 1275 1276 1363 1413 1459 1505 1551 1597 1657 1691 1731 1762
1794 1827 1872 1917 1962 2007 2052 2097 2142 2187 2232 2277 2322
2367 2412 2457 2502 2547 2592 2637 2682 2727 2772 2817 2862 2907
2952 2997 3042 3089 3135 3181 3227 3273 3319 3365 3411 3457 3502
3547 3592 3637 3682 3727 3772 3817 3874 3923 3979 4029 4078 4115
4158 4202 4240 4283 4325 4361 4396 4432 4467 4499 4535 4584 4649
4698 4755 4794 4833 4872 4911 4950 4990 5034 5076 5113 5160 5202
5250 5287 5322 5353 5388 5432 5504 5554 5602 5668 5717 5744 5785
5827 5869 5911 5953 5995 6038 6130 6228 6296 6350 6392 6432 6524
6621 6691 6744 6856 6907 6985 7063 7141 7221 7326 7431 7536 7641
7750 7832 7914 7996 8079 8182 8285 8388 8491 8646 8654 8666 8672
8674 9234 9235 9236 9237 9238 9246 9253 9265 9268 9276 9399 9400
9401 9402 9403 9408 9420 9422 9423 9426 9427 9428 9435 9436 9437
9449 9452 9455 9747

$SWREG 001222 987# 1297
$SWRMK= 000000 742 743 9403 9404 9424
$TESTN 001204 978# 9447*
$TIMES 001164 963# 1272* 1363* 1657* 1691* 1731* 1762* 1794* 1827* 1872* 1917* 1962* 2007*
2052* 2097* 2142* 2187* 2232* 2277* 2322* 2367* 2412* 2457* 2502* 2547* 2592*
2637* 2682* 2727* 2772* 2817* 2862* 2907* 2952* 2997* 3042* 3089* 3135* 3181*
3227* 3273* 3319* 3365* 3411* 3457* 3502* 3547* 3592* 3637* 3682* 3727* 3772*
3817* 3874* 3923* 3979* 4029* 4078* 4115* 4158* 4202* 4240* 4283* 4325* 4361*
4396* 4432* 4584* 4755* 4794* 4833* 4872* 4911* 4950* 4990* 5287* 5504* 5602*
5717* 5744* 5785* 5827* 5869* 5911* 5953* 5995* 6296* 6691* 6907* 6985* 7063*
7141* 7221* 7326* 7431* 7536* 7641* 7750* 7832* 7914* 7996* 8079* 8182* 8285*
8388* 8491* 8654* 9435* 9442 9445* 9455

$TKB 001144 952# 9460 9468 9527
$TKS 001142 951# 9460 9466 9525
$TMPO 001162 962# 1380* 1426* 1472* 1518* 1564* 1610* 7273* 7275 7292 7378* 7380 7397
7483* 7485 7502 7588* 7590 7607 7693* 7695 7712 8130* 8132 8149 8233*
8235 8252 8336* 8338 8355 8439* 8441 8458 8542* 8544 8561 9964 9966
9968 9970 9972

$TN = 000211 730# 1350 1363# 1400 1413# 1446 1459# 1492 1505# 1538 1551# 1584 1597#
1631 1657# 1677 1691# 1711 1731# 1751 1762# 1784 1794# 1816 1827# 1861
1872# 1906 1917# 1951 1962# 1996 2007# 2041 2052# 2086 2097# 2131 2142#
2176 2187# 2221 2232# 2266 2277# 2311 2322# 2356 2367# 2401 2412# 2446
2457# 2491 2502# 2536 2547# 2581 2592# 2626 2637# 2671 2682# 2716 2727#
2761 2772# 2806 2817# 2851 2862# 2896 2907# 2941 2952# 2986 2997# 3031
3042# 3078 3089# 3124 3135# 3170 3181# 3216 3227# 3262 3273# 3308 3319#
3354 3365# 3400 3411# 3446 3457# 3491 3502# 3536 3547# 3581 3592# 3626
3637# 3671 3682# 3716 3727# 3761 3772# 3806 3817# 3856 3874# 3905 3923#
3961 3979# 4011 4029# 4065 4078# 4102 4115# 4145 4158# 4189 4202# 4227
4240# 4270 4283# 4314 4325# 4349 4361# 4385 4396# 4420 4432# 4457 4467#
4488 4499# 4520 4535# 4564 4584# 4636 4649# 4685 4698# 4744 4755# 4783
4794# 4822 4833# 4861 4872# 4900 4911# 4939 4950# 4978 4990# 5021 5034#
5064 5076# 5098 5113# 5145 5160# 5182 5202# 5230 5250# 5279 5287# 5311
5322# 5342 5353# 5374 5388# 5409 5432# 5491 5504# 5535 5554# 5585 5602#
5654 5668# 5706 5717# 5740 5744# 5774 5785# 5816 5827# 5858 5869# 5900
5911# 5942 5953# 5984 5995# 6026 6038# 6084 6130# 6131 6212 6228# 6280
6296# 6337 6350# 6381 6392# 6421 6432# 6478 6524# 6525 6606 6621# 6675
6691# 6731 6744# 6836 6856# 6890 6907# 6968 6985# 7046 7063# 7124 7141#
7207 7221# 7312 7326# 7417 7431# 7522 7536# 7627 7641# 7732 7750# 7814
7832# 7896 7914# 7978 7996# 8065 8079# 8166 8182# 8271 8285# 8374 8388#
8477 8491#

```
$TPB    001150            954#  9641*  9652
$TPFLG  001155            958#  9590   9652
$TPS    001146            953#  9639   9652
$TRAP   027312           1268   9763#
$TRP  = 000011           9771#  9781#  9782#  9783#  9784#  9785#  9786#  9787#  9788#  9789#
$TRPAD  027334           9768   9779#
$TSTM   001004            918#
$TSTNM  001102            933#  1365*  8653*  9245   9276   9398   9424   9446*  9447   9452   9456
$TTYIN  026340           9543   9544   9561   9565#
$TYPBN= ****** U         9785
$TYPOS  025210           9337#  9784
$TYPE   026404           9590#  9684   9771   9780
$TYPEC  026616           9505   9620   9627   9634   9639#  9640
$TYPEX  026664           9645   9647   9650#
$TYPOC  024470           9180#  9781
$TYPON  024504           9179   9182#  9783
$TYPOS  024444           9175#  9782
$UNIT   001212            981#
$UNITM  001010            920#
$USWR   001224            988#
$VECT1  001250           1013#  1308   8634
$VECT2  001251           1014#
$XTSTR  025444           9411#
$$GET4= 000000           8666#
$OFILL  024667           9176*  9180*  9190   9225#
$40CAT= ****** U         9255   9408
.     = 031264            746#   750#   753#   892    893#   895#   897#   898#   905    906#   908#   910#   930#
                          969   1261   1275   1276   1305#  6144#  6538#  8598#  8625#  8674   8677   8761#  8843#
                         8926#  9009#  9092#  9115#  9276   9323#  9391#  9455   9456   9460   9565#  9566   9572
                         9652   9705#  9727   9749   9946#
.$ASTA= ****** U         9657   9660
.$X   = 001000            905#   910
```

DZKWK.CMB CROSS REFERENCE TABLE -- MACRO NAMES

ADTST 1347# 1348 1399 1445 1491 1537 1583
AREPT 7202# 7203 7308 7413 7518 7623
BREPT 8060# 8061 8164 8267 8370 8473
BRTM 5771# 5772 5814 5856 5898 5940 5982
CADT 6887# 6888 6966 7044 7122
CADTB 7729# 7730 7812 7894 7976
CBC 3075# 3076 3122 3168 3214 3260 3306 3352 3398 3444
CBTC 5310# 5341
CBT2S 5181# 5229
COMMEN 874# 1383 1394 1429 1440 1475 1486 1521 1532 1567 1578 1613 1624 1666 1672
1699 1705 1738 1745 1770 1777 1802 1809 1835 1841 1850 1856 1880 1886 1895
1901 1925 1931 1940 1946 1970 1976 1985 1991 2015 2021 2030 2036 2060 2066
2075 2081 2105 2111 2120 2126 2150 2156 2165 2171 2195 2201 2210 2216 2240
2246 2255 2261 2285 2291 2300 2306 2330 2336 2345 2351 2375 2381 2390 2396
2420 2426 2435 2441 2465 2471 2480 2486 2510 2516 2525 2531 2555 2561 2570
2576 2600 2606 2615 2621 2645 2651 2660 2666 2690 2696 2705 2711 2735 2741
2750 2756 2780 2786 2795 2801 2825 2831 2840 2846 2870 2876 2885 2891 2915
2921 2930 2936 2960 2966 2975 2981 3005 3011 3020 3026 3050 3056 3065 3071
3097 3103 3112 3118 3143 3149 3158 3164 3189 3195 3204 3210 3235 3241 3250
3256 3281 3287 3296 3302 3327 3333 3342 3348 3373 3379 3388 3394 3419 3425
3434 3440 3465 3471 3480 3486 3510 3516 3525 3531 3555 3561 3570 3576 3600
3606 3615 3621 3645 3651 3660 3666 3690 3696 3705 3711 3735 3741 3750 3756
3780 3786 3795 3801 3825 3831 3840 3846 3892 3900 3946 3954 3997 4005 4051
4059 4090 4097 4132 4140 4175 4183 4214 4221 4257 4265 4300 4308 4337 4344
4373 4380 4408 4415 4444 4451 4475 4482 4506 4514 4548 4558 4610 4617 4658
4664 4673 4681 4711 4717 4726 4737 4771 4777 4810 4816 4849 4855 4888 4894
4927 4933 4966 4972 5007 5015 5051 5058 5086 5092 5124 5139 5171 5176 5216
5224 5264 5272 5299 5305 5330 5337 5361 5368 5397 5403 5456 5466 5477 5486
5520 5526 5563 5579 5631 5638 5680 5686 5694 5700 5730 5736 5759 5766 5804
5810 5846 5852 5888 5894 5930 5936 5972 5978 6014 6020 6056 6070 6171 6178
6199 6206 6260 6268 6321 6330 6449 6463 6565 6572 6593 6600 6653 6661 6716
6725 6761 6772 6801 6822 6871 6881 6934 6940 6955 6962 7012 7018 7033 7040
7090 7096 7111 7118 7168 7174 7189 7196 7262 7268 7280 7287 7296 7303 7367
7373 7385 7392 7401 7408 7472 7478 7490 7497 7506 7513 7577 7583 7595 7602
7611 7618 7682 7688 7700 7707 7716 7723 7778 7785 7801 7808 7860 7867 7883
7890 7942 7949 7965 7972 8024 8031 8047 8054 8119 8125 8137 8144 8153 8160
8222 8228 8240 8247 8256 8263 8325 8331 8343 8350 8359 8366 8428 8434 8446
8453 8462 8469 8531 8537 8549 8556 8565 8572 8732 8739 8815 8821 8898 8904
8980 8987 9063 3070 9131 9140
CSRDTA 1815# 1860 1905 1950 1995 2040 2085 2130 2175 2220 2265 2310 2355 2400 2445
2490 2535 2580 2625 2670 2715 2760 2805 2850 2895 2940 2985 3030 3077 3123
3169 3215 3261 3307 3353 3399 3445 3490 3535 3580 3625 3670 3715 3760 3805
DFC 1077# 1083 1091 1099 1107 1115 1123 1131 1140 1148 1156 1171 1179 1187 1202
1210 1218 1226 1234 1242 1250
DUMC 6080# 6229 6622
DUMWRN 6078# 6131 6525
ECALL 1815#
ECB 884# 1382 1393 1428 1439 1474 1485 1520 1531 1566 1577 1612 1623 1665 1671
1698 1704 1737 1744 1769 1776 1801 1808 1834 1840 1849 1855 1879 1885 1894
1900 1924 1930 1939 1945 1969 1975 1984 1990 2014 2020 2029 2035 2059 2065
2074 2080 2104 2110 2119 2125 2149 2155 2164 2170 2194 2200 2209 2215 2239
2245 2254 2260 2284 2290 2299 2305 2329 2335 2344 2350 2374 2380 2389 2395
2419 2425 2434 2440 2464 2470 2479 2485 2509 2515 2524 2530 2554 2560 2569
2575 2599 2605 2614 2620 2644 2650 2659 2665 2689 2695 2704 2710 2734 2740
2749 2755 2779 2785 2794 2800 2824 2830 2839 2845 2869 2875 2884 2890 2914
2920 2929 2935 2959 2965 2974 2980 3004 3010 3019 3025 3049 3055 3064 3070

```
                3096  3102  3111  3117  3142  3148  3157  3163  3188  3194  3203  3209  3234  3240  3249
                3255  3280  3286  3295  3301  3326  3332  3341  3347  3372  3378  3387  3393  3418  3424
                3433  3439  3464  3470  3479  3485  3509  3515  3524  3530  3554  3560  3569  3575  3598
                3605  3614  3620  3644  3650  3659  3665  3689  3695  3704  3710  3734  3740  3749  3755
                3779  3785  3794  3800  3824  3830  3839  3845  3891  3899  3945  3953  3996  4004  4050
                4058  4089  4096  4131  4139  4174  4182  4213  4220  4256  4264  4299  4307  4336  4343
                4372  4379  4407  4414  4443  4450  4474  4481  4505  4513  4547  4557  4609  4616  4657
                4663  4672  4680  4710  4716  4725  4736  4770  4776  4809  4815  4848  4854  4887  4893
                4926  4932  4965  4971  5006  5014  5050  5057  5085  5091  5123  5138  5170  5175  5215
                5223  5263  5271  5298  5304  5329  5336  5360  5367  5396  5402  5455  5465  5476  5485
                5519  5525  5562  5578  5630  5637  5679  5685  5693  5699  5729  5735  5758  5765  5803
                5809  5845  5851  5887  5893  5929  5935  5971  5977  6013  6019  6055  6069  6170  6177
                6198  6205  6259  6267  6320  6329  6448  6462  6564  6571  6592  6599  6652  6660  6715
                6724  6760  6771  6800  6821  6870  6880  6933  6939  6954  6961  7011  7017  7032  7039
                7089  7095  7110  7117  7167  7173  7188  7195  7261  7267  7279  7286  7295  7302  7366
                7372  7384  7391  7400  7407  7471  7477  7489  7496  7505  7512  7576  7582  7594  7601
                7610  7617  7681  7687  7699  7706  7715  7722  7777  7784  7800  7807  7859  7866  7882
                7889  7941  7948  7964  7971  8023  8030  8046  8053  8118  8124  8136  8143  8152  8159
                8221  8227  8239  8246  8255  8262  8324  8330  8342  8349  8358  8365  8427  8433  8445
                8452  8461  8468  8530  8536  8548  8555  8564  8571  8731  8738  8814  8820  8897  8903
                8979  8986  9062  9069  9130  9139
ENDCOM           874#  1386  1397  1432  1443  1478  1489  1524  1535  1570  1581  1616  1627  1669  1675
                1702  1708  1741  1748  1773  1780  1805  1812  1838  1844  1853  1859  1883  1889  1898
                1904  1928  1934  1943  1949  1973  1979  1988  1994  2018  2024  2033  2039  2063  2069
                2078  2084  2108  2114  2123  2129  2153  2159  2168  2174  2198  2204  2213  2219  2243
                2249  2258  2264  2288  2294  2303  2309  2333  2339  2348  2354  2378  2384  2393  2399
                2423  2429  2438  2444  2468  2474  2483  2489  2513  2519  2528  2534  2558  2564  2573
                2579  2603  2609  2618  2624  2648  2654  2663  2669  2693  2699  2708  2714  2738  2744
                2753  2759  2783  2789  2798  2804  2828  2834  2843  2849  2873  2879  2888  2894  2918
                2924  2933  2939  2963  2969  2978  2984  3008  3014  3023  3029  3053  3059  3068  3074
                3100  3106  3115  3121  3146  3152  3161  3167  3192  3198  3207  3213  3238  3244  3253
                3259  3284  3290  3299  3305  3330  3336  3345  3351  3376  3382  3391  3397  3422  3428
                3437  3443  3468  3474  3483  3489  3513  3519  3528  3534  3558  3564  3573  3579  3603
                3609  3618  3624  3648  3654  3663  3669  3693  3699  3708  3714  3738  3744  3753  3759
                3783  3789  3798  3804  3828  3834  3843  3849  3895  3903  3949  3957  4000  4008  4054
                4062  4093  4100  4135  4143  4178  4186  4217  4224  4260  4268  4303  4311  4340  4347
                4376  4383  4411  4418  4447  4454  4478  4485  4509  4517  4551  4561  4613  4620  4661
                4667  4676  4684  4714  4720  4729  4740  4774  4780  4813  4819  4852  4858  4891  4897
                4930  4936  4969  4975  5010  5018  5054  5061  5089  5095  5127  5142  5174  5179  5219
                5227  5267  5275  5302  5308  5333  5340  5364  5371  5400  5406  5459  5469  5480  5489
                5523  5529  5566  5582  5634  5641  5683  5689  5697  5703  5733  5739  5762  5769  5807
                5813  5849  5855  5891  5897  5933  5939  5975  5981  6017  6023  6059  6073  6174  6181
                6202  6209  6263  6271  6324  6333  6452  6466  6568  6575  6596  6603  6656  6664  6719
                6728  6764  6775  6804  6825  6874  6884  6937  6943  6958  6965  7015  7021  7036  7043
                7093  7099  7114  7121  7171  7177  7192  7199  7265  7271  7283  7290  7299  7306  7370
                7376  7388  7395  7404  7411  7475  7481  7493  7500  7509  7516  7580  7586  7598  7605
                7614  7621  7685  7691  7703  7710  7719  7726  7781  7788  7804  7811  7863  7870  7886
                7893  7945  7952  7968  7975  8027  8034  8050  8057  8122  8128  8140  8147  8156  8163
                8225  8231  8243  8250  8259  8266  8328  8334  8346  8353  8362  8369  8431  8437  8449
                8456  8465  8472  8534  8540  8552  8559  8568  8575  8735  8742  8818  8824  8901  8907
                8983  8990  9066  9073  9099  9134  9143
ERROR            768#  1386  1432  1478  1524  1570  1616  1669  1702  1741  1773  1805  1838  1853  1883
                1898  1928  1943  1973  1988  2018  2033  2063  2078  2108  2123  2153  2168  2198  2213
                2243  2258  2288  2303  2333  2348  2378  2393  2423  2438  2468  2483  2513  2528  2558
                2573  2603  2518  2648  2663  2693  2708  2738  2753  2783  2798  2828  2843  2873  2888
                2918  2933  2963  2978  3008  3023  3053  3068  3100  3115  3146  3161  3192  3207  3238
```

```
                3253  3284  3299  3330  3345  3376  3391  3422  3437  3468  3483  3513  3528  3558  3573
                3603  3618  3648  3663  3693  3708  3738  3753  3783  3798  3828  3843  3895  3949  4000
                4054  4093  4135  4178  4217  4260  4303  4340  4376  4411  4447  4478  4509  4551  4613
                4661  4676  4714  4729  4774  4813  4852  4891  4930  4969  5010  5054  5089  5127  5174
                5219  5267  5302  5333  5364  5400  5459  5480  5523  5566  5634  5683  5697  5733  5762
                5807  5849  5891  5933  5975  6017  6059  6174  6202  6263  6324  6365  6408  6452  6568
                6596  6656  6719  6764  6804  6875  6937  6958  7015  7036  7093  7114  7171  7192  7265
                7283  7299  7370  7388  7404  7475  7493  7509  7580  7598  7614  7685  7703  7719  7781
                7804  7863  7886  7945  7968  8027  8050  8123  8140  8156  8226  8243  8259  8329  8346
                8362  8432  8449  8465  8535  8552  8568  8736  8818  8901  8983  9066
ESCAPE           874#
GETPRI           874#
ISEXP           6082#  6105  6499
MSY              882#  1356  1406  1452  1498  1544  1590  1650  1684  1724  1755  1788  1821  1866  1911
                1956  2001  2046  2091  2136  2181  2226  2271  2316  2361  2406  2451  2496  2541  2586
                2631  2676  2721  2766  2811  2856  2901  2946  2991  3036  3083  3129  3175  3221  3267
                3313  3359  3405  3451  3496  3541  3586  3631  3676  3721  3766  3811  3865  3914  3970
                4023  4071  4108  4151  4195  4233  4276  4319  4356  4391  4427  4461  4493  4529  4578
                4644  4692  4749  4788  4827  4866  4905  4944  4985  5029  5071  5106  5154  5196  5244
                5316  5347  5382  5427  5499  5548  5597  5662  5711  5779  5821  5863  5905  5947  5989
                6032  6102  6223  6287  6344  6386  6426  6496  6616  6685  6738  6851  6902  6980  7058
                7136  7216  7321  7426  7531  7636  7745  7827  7909  7991  8074  8177  8280  8383  8486
                8700  8783  8865  8945  9028
MTAGS            880#  1042
MULT             874#
NEWTST           874#  1350  1400  1446  1492  1538  1584  1631  1677  1711  1751  1784  1816  1861  1906
                1951  1996  2041  2086  2131  2176  2221  2266  2311  2356  2401  2446  2491  2536  2581
                2626  2671  2716  2761  2806  2851  2896  2941  2986  3031  3078  3124  3170  3216  3262
                3308  3354  3400  3446  3491  3536  3581  3626  3671  3716  3761  3806  3856  3905  3961
                4011  4065  4102  4145  4189  4227  4270  4314  4349  4385  4420  4457  4488  4520  4564
                4636  4685  4744  4783  4822  4861  4900  4939  4978  5021  5064  5098  5145  5182  5230
                5279  5311  5342  5374  5409  5491  5535  5585  5654  5706  5740  5774  5816  5858  5900
                5942  5984  6026  6084  6212  6280  6337  6381  6421  6478  6606  6675  6731  6836  6890
                6968  7046  7124  7207  7312  7417  7522  7627  7732  7814  7896  7978  8065  8168  8271
                8374  8477
POP              874#  9378  9699  9700  9736  9737
POFSP2           881#  1378  1424  1470  1516  1562  1608  6185  6196  6274  6317  6374  6415  6470  6579
                6590  6668  6712  6798  8616
PUSH             874#  9337  9660  9662  9683  9717  9723
REPORT           874#
RTM             4741#  4742  4781  4820  4859  4898  4937
SCOPE            769#  1412  1458  1504  1550  1596  1656  1690  1730  1761  1793  1826  1871  1916  1961
                2006  2051  2096  2141  2186  2231  2276  2321  2366  2411  2456  2501  2546  2591  2635
                2681  2726  2771  2816  2861  2906  2951  2996  3041  3088  3134  3180  3226  3272  3318
                3364  3410  3456  3501  3546  3591  3636  3681  3726  3771  3816  3873  3922  3978  4028
                4077  4114  4157  4201  4239  4282  4324  4360  4395  4431  4466  4498  4534  4583  4648
                4697  4754  4793  4832  4871  4910  4949  4989  5033  5075  5112  5159  5201  5249  5286
                5321  5352  5387  5431  5503  5553  5601  5667  5716  5743  5784  5826  5868  5910  5952
                5994  6037  6129  6227  6295  6349  6391  6431  6523  6620  6690  6743  6855  6906  6984
                7062  7140  7220  7325  7430  7535  7640  7749  7831  7913  7995  8078  8181  8284  8387
                8490  8584
SETPRI           874#
SETTRA          9771#  9781  9782  9783  9784  9785  9786  9787  9788
SETUP            874#  1256
SKIP             874#
SLASH            874#
```

```
SPACE    874#
STARS    874#   890   902   904   911   926   969   973  1350  1361  1400  1411  1446  1457  1492
        1503  1538  1549  1584  1595  1631  1655  1677  1689  1711  1729  1751  1760  1784  1795
        1816  1825  1861  1870  1906  1915  1951  1960  1996  2005  2041  2050  2086  2095  2131
        2140  2176  2185  2221  2230  2266  2275  2311  2320  2356  2365  2401  2410  2446  2455
        2491  2500  2536  2545  2581  2590  2626  2635  2671  2680  2716  2725  2761  2770  2806
        2815  2851  2860  2896  2905  2941  2950  2986  2995  3031  3040  3078  3087  3124  3133
        3170  3179  3216  3225  3262  3271  3308  3317  3354  3363  3400  3409  3446  3455  3491
        3500  3536  3545  3581  3590  3626  3635  3671  3680  3716  3725  3761  3770  3806  3815
        3856  3872  3905  3921  3961  3977  4011  4027  4065  4076  4102  4113  4145  4156  4189
        4200  4227  4238  4270  4281  4314  4323  4349  4359  4385  4394  4420  4430  4457  4465
        4488  4497  4520  4533  4564  4582  4636  4647  4685  4696  4744  4753  4783  4792  4822
        4831  4861  4870  4900  4909  4939  4948  4978  4988  5021  5032  5064  5074  5098  5111
        5145  5158  5182  5200  5230  5248  5279  5285  5311  5320  5342  5351  5374  5386  5409
        5430  5491  5502  5535  5552  5585  5600  5654  5666  5706  5715  5740  5742  5774  5783
        5816  5825  5858  5867  5900  5909  5942  5951  5984  5993  6026  6036  6084  6128  6212
        6226  6280  6294  6337  6348  6381  6390  6421  6430  6478  6522  6606  6619  6675  6689
        6731  6742  6836  6854  6890  6905  6968  6983  7046  7061  7124  7139  7207  7219  7312
        7324  7417  7429  7522  7534  7627  7639  7732  7748  7814  7830  7896  7912  7978  7994
        8065  8077  8168  8180  8271  8283  8374  8386  8477  8489  8643  9152  9230  9279  9327
        9395  9459  9515  9535  9575  9655  9713  9729  9757
STM1     8682#  8710  8792  8875  8954  9037
STM2     8683#  8743  8825  8908  8991  9074
SWRSU     874#  1277#
TRMTRP   9771#
TYPBIN    874#
TYPDEC    874#
TYPNAM    874#
TYPNUM    874#
TYPOCS    719#   874#
TYPOCT    874#  8599  8610  8626  9109  9121  9291  9315  9475
TYPTXT    874#  1302  6141  6236  6535  6629  8595  8601  8606  8622  8628  8751  8758  8833  8840
         8916  8923  8999  9006  9082  9089  9104  9112
UPDATE    886#
ZP21     3854#  3858  3907  3963
ZP22     3959#  4013
ZP23     4064#  4067  4104  4147
ZP24     4188#  4191  4229  4272
ZP25     4313#  4316
ZP25A    4348#  4351
ZP25B    4384#  4387
ZP25C    4419#  4422
ZP26     4456#  4459
ZP27     4487#  4490
ZP28     4519#  4522
ZP29     4563#  4566
ZP3E     4743#  4746  4782#  4785  4821#  4824  4860#  4863  4899#  4902  4938#  4941
ZP3EA    4977#  4980
ZP30     4635#  4638
ZP31     4685#  4687
ZP310    5373#  5376
ZP311    5408#  5411
ZP312    5490#  5493
ZP33     5020#  5023
ZP34     5063#  5066
ZP35     5097#  5100
```

```
ZP36    5144#  5147
ZP37    5181#  5184   5229#  5232
ZP38    5278#  5281
ZP39    5310#  5313   5341#  5344
ZP42    5534#  5537
ZP43    5584#  5587
ZP44    5653#  5656
ZP45    5705#  5708
ZP46E   5772#  5776   5814#  5818   5856#  5860   5898#  5902   5940#  5944   5982#  5986
ZP47    6025#  6028
ZP50    6083#  6086
ZP51    6211#  6214
ZP52    6279#  6282
ZP53    6336#  6339
ZP54    6380#  6383
ZP55    6420#  6423
ZP56    6477#  6480
ZP57    6605#  6608
ZP58    6674#  6677
ZP59    6730#  6733
ZP6E    6834#  6838
ZP61E   6888#  6892   6966#  6970   7044#  7048   7122#  7126   7730#  7734   7812#  7816   7894#  7898   7976#
        7980
ZP64E   7205#  7209   7310#  7314   7415#  7419   7520#  7524   7625#  7629   8063#  8067   8166#  8170   8269#
        8273   8372#  8376   8475#  8479
ZZ1     1814#  1818   1863   1908   1953   1998   2043   2088   2133   2178   2223   2268   2313   2358   2403
        2448   2493   2538   2583   2628   2673   2718   2763   2808   2853   2898   2943   2988   3033   3080
        3126   3172   3218   3264   3310   3356   3402   3448   3493   3538   3583   3628   3673   3718   3763
        3808   4068   4105   4148   4192   4230   4273
Z1      1349#  1352   1400#  1402   1446#  1448   1492#  1494   1538#  1540   1584#  1586
Z2      1630#  1633
Z3      1676#  1679
Z4      1710#  1713
Z6      1750#  1753
Z7      1783#  1786
$SCMRE   923#   961
$SCMTM   923#   962
$SESCA   874#
$SNEWT   874#  1350   1400   1446   1492   1538   1584   1631   1677   1711   1751   1784   1816   1861   1906
        1951   1996   2041   2086   2131   2176   2221   2266   2311   2356   2401   2446   2491   2536   2581
        2626   2671   2716   2761   2806   2851   2896   2941   2986   3031   3078   3124   3170   3216   3262
        3308   3354   3400   3446   3491   3536   3581   3626   3671   3716   3761   3806   3856   3905   3961
        4011   4065   4102   4145   4189   4227   4270   4314   4349   4385   4420   4457   4488   4520   4564
        4636   4685   4744   4783   4822   4861   4900   4939   4978   5021   5064   5098   5145   5182   5230
        5279   5311   5342   5374   5409   5491   5535   5585   5654   5706   5740   5774   5816   5858   5900
        5942   5984   6026   6094   6212   6280   6337   6381   6421   6478   6606   6675   6731   6836   6890
        6968   7046   7124   7207   7312   7417   7522   7627   7732   7814   7896   7978   8065   8168   8271
        8374   8477
$SSET   9771#  9781   9782   9783   9784   9785   9786   9787   9788
$SSETM  1293#  1294
$SSKIP   874#
.EQUAT   719#   763
.HEADE   719#   720
.SETTR   719#
.SETUP   719#  1255
.SWRHI   719#   731
```

.SWRLO	743#	
.TRMTR	719#	
.SACT1	719#	887
.SAPTB	719#	970#
.SAPTH	719#	899
.SAPTY	719#	9652
.SCATC	719#	743
.SCMTA	719#	923
.SEOP	719#	8640
.SERRO	719#	9227
.SERRT	719#	9276
.SPOWE	719#	9710
.SRDOC	719#	
.SREAD	719#	9456
.SSCOP	719#	9392
.STRAP	719#	9754
.STYPD	719#	9324
.STYPE	719#	9572
.STYPO	719#	9149

```
ADD     1318  1320  1322  1324  1326  1330  1332  1334  1379  1425  1471  1517  1563  1609  6186
        6197  6275  6318  6375  6416  6471  6580  6591  6669  6713  6799  8588  8613  8617  9117
        9178  9188  9299  9357  9485  9493  9500  9502  9609  9670  9682  9694
ASL     9296  9297  9298  9509  9510  9511  9767
ASLB    9362
ASR     6245  6246  6247  6301  6302  6303  6638  6639  6640  6696  6697  6698  9677
BCC     9363
BEQ     1296  1696  1833  1848  1878  1893  1923  1938  1968  1983  2013  2028  2058  2073  2103
        2118  2148  2163  2193  2208  2238  2253  2283  2298  2328  2343  2373  2388  2418  2433
        2463  2478  2508  2523  2553  2568  2598  2613  2643  2658  2688  2703  2733  2748  2778
        2793  2823  2838  2868  2883  2913  2928  2958  2973  3003  3018  3048  3063  3095  3110
        3141  3156  3187  3202  3233  3248  3279  3294  3325  3340  3371  3386  3417  3432  3463
        3478  3508  3523  3553  3568  3598  3613  3643  3658  3688  3703  3733  3748  3778  3793
        3823  3838  3889  3943  3994  4048  4087  4129  4172  4211  4254  4297  4334  4370  4405
        4441  4545  4607  4622  4625  4709  4724  5004  5049  5122  5214  5262  5297  5395  5454
        5475  5629  5644  5647  5692  5728  5757  6868  6932  6952  7010  7030  7088  7108  7166
        7186  7776  7798  7858  7880  7940  7962  8022  8044  8620  8665  8813  8969  9052  9125
        9127  9205  9244  9247  9269  9272  9301  9306  9319  9423  9425  9427  9431  9440  9491
        9498  9508  9599  9612  9647  9664  9668  9688  9690
BGE     9443
BGT     8659  9212  9371  9532
BHI     9429
BIC     1845  1890  1935  1980  2025  2070  2115  2160  2205  2250  2295  2340  2385  2430  2475
        2520  2565  2610  2655  2700  2745  2790  2835  2880  2925  2970  3015  3060  3107  3153
        3199  3245  3291  3337  3383  3429  3475  3520  3565  3610  3655  3700  3745  3790  3835
        6248  6304  6641  6699  8092  8104  8195  8207  8298  8310  8401  8413  8504  8516  8656
        9202  9469  9506  9528  9533
BIS     1829  1874  1919  1964  2009  2054  2099  2144  2189  2234  2279  2324  2369  2414  2459
        2504  2549  2594  2639  2684  2729  2774  2819  2864  2909  2954  2999  3044  3091  3137
        3183  3229  3275  3321  3367  3413  3459  3504  3549  3594  3639  3684  3729  3774  3819
        4469  4501  4540  4598  4653  4702  5080  5117  5164  5209  5257  5326  5357  5391  5438
        5558  5559  5620  5675  6048  6165  6166  6254  6255  6309  6310  6354  6355  6397  6398
        6438  6558  6559  6647  6648  6704  6705  6750  6859  6913  6917  6946  6991  6995  7024
        7069  7073  7102  7147  7151  7180  7756  7760  7792  7838  7842  7874  7920  7924  7956
        8002  8006  8038  8721  8804  8885  9207  9208  9365  9366  9513
BISB    9288
BIT     4502  4624  4655  4669  4708  4766  4805  4844  4883  4922  4961  5002  5083  5168  5646
        8727  8744  8826  8909  8968  8992  9051  9075  9246  9253  9268  9408  9422  9430  9437
BITB    1295  9598  9603  9635  9667
BLE     7260  7278  7294  7365  7383  7399  7470  7488  7504  7575  7593  7609  7680  7698  7714
        8117  8135  8151  8220  8238  8254  8323  8341  8357  8426  8444  8460  8529  8547  8563
BLOS    9545
BLT     9213  9354  9370  9530  9626
BMI     4472  5678  5802  5844  5886  5928  5970  6012  8586  8757  8839  8896  8922  8978  9005
        9061  9088  9361
BNE     1261  1284  1300  1316  1662  1736  1767  1799  4503  4656  4671  4762  4764  4769  4801
        4803  4808  4840  4842  4847  4879  4881  4886  4918  4920  4925  4957  4959  4964  4998
        5000  5084  5169  5328  5359  5516  5561  5753  5794  5797  5836  5839  5878  5881  5920
        5923  5962  5965  6004  6007  6054  6134  6136  6231  6233  6528  6530  6624  6626  6919
        6925  6948  6997  7003  7026  7075  7081  7104  7153  7159  7182  7232  7244  7337  7349
        7442  7454  7547  7559  7652  7664  7762  7769  7794  7844  7851  7876  7926  7933  7958
        8008  8015  8040  8090  8102  8193  8205  8296  8308  8399  8411  8502  8514  8729  8745
        8748  8827  8830  8910  8913  8993  8996  9076  9079  9203  9254  9259  9289  9311  9359
        9409  9438  9465  9471  9483  9489  9496  9549  9555  9597  9604  9606  9614  9622  963C
        9643  9666  9672  9675  9692  9735
BPL     5512  5518  7253  7358  7463  7568  7673  8111  8214  8317  8420  8523  9201  9266  9345
```

```
          9375  9467  9526  9591  9640
BR        1303  1377  1423  1469  1515  1561  1607  1844  1885  1934  1979  2024  2069  2114  2159
          2204  2249  2294  2339  2384  2429  2474  2519  2564  2609  2654  2699  2744  2789  2834
          2879  2924  2969  3014  3059  3106  3152  3198  3244  3290  3336  3282  3428  3474  3519
          3564  3609  3654  3699  3744  3789  3834  4620  4627  4667  4720  5464  5641  5649  5689
          6142  6181  6194  6237  6271  6313  6369  6411  6466  6536  6575  6588  6630  6664  6708
          6775  6796  6943  7021  7099  7177  7271  7290  7376  7395  7481  7500  7586  7605  7691
          7710  7788  7870  7952  8034  8128  8147  8231  8250  8334  8353  8437  8456  8540  8559
          8596  8602  8607  8623  8629  8752  8759  8762  8834  8841  8844  8917  8924  8927  9000
          9007  9010  9083  9090  9093  9105  9113  9135  9179  9194  9215  9264  9294  9321  9356
          9373  9411  9417  9420  9433  9436  9486  9494  9514  9551  9593  9619  9629  9638  9645
          9658  9680  9727  9749
CLR       1259  1272  1273  1294  1311  1314  1368  1369  1414  1415  1460  1461  1506  1507  1552
          1553  1598  1599  1693  1694  1748  1828  1846  1873  1891  1918  1936  1963  1981  2008
          2026  2053  2071  2098  2116  2143  2161  2188  2206  2233  2251  2278  2296  2323  2341
          2368  2386  2413  2431  2458  2476  2503  2521  2548  2566  2593  2611  2638  2656  2683
          2701  2728  2746  2773  2791  2818  2836  2863  2881  2908  2926  2953  2971  2998  3016
          3043  3061  3090  3108  3136  3154  3182  3200  3228  3246  3274  3292  3320  3338  3366
          3384  3412  3430  3458  3476  3503  3521  3548  3566  3593  3611  3638  3656  3683  3701
          3728  3746  3773  3791  3818  3836  3876  3925  3981  4030  4080  4081  4085  4117  4160
          4204  4205  4209  4242  4285  4327  4363  4398  4434  4468  4517  4536  4537  4590  4650
          4699  4757  4758  4760  4796  4797        4835  4836  4838  4874  4875  4877  4913  4914
          4916  4952  4953  4955  4992  4993        5036  5037  5077  5114  5162  5204  5208  5252
          5256  5289  5292  5323  5354  5433        5489  5506  5508  5510  5529  5556  5609  5669
          5670  5719  5720  5725  5746  5747        5751  5787  5788  5789  5792  5829  5830  5831
          5834  5871  5872  5873  5876  5913        5915  5918  5955  5956  5957  5960  5997  5998
          5999  6002  6039  6040  6041  6154  6155  6160  6189  6334  6351  6352  6358  6393  6394
          6402  6417  6433  6434  6442  6473  6548  6549  6553  6583  6672  6745  6746  6751  6781
          6790  6858  6909  6910  6911  6987  6988  6989  7065  7066  7067  7143  7144  7145  7223
          7224  7225  7234  7237  7246  7328  7329  7330  7339  7342  7351  7433  7434  7435  7444
          7447  7456  7538  7539  7540  7549  7552  7561  7643  7644  7645  7654  7657  7666  7752
          7753  7754  7834  7835  7836  7916  7917  7918  7998  7999  8000  8081  8082  8083  8095
          8163  8184  8185  8186  8198  8266  8287  8288  8289  8301  8369  8390  8391  8392  8404
          8472  8493  8494  8495  8507  8575  8653  8654  8711  8717  8718  8793  8799  8800  8876
          8882  8883  8955  8961  8962  9038  9044  9045  9192  9287  9348  9351  9435  9450  9478
          9479  9733
CLRB       3885  3941  3990  4046  9377  9434  9556  9618  9644  9696  9697  9698
CMP        1260  1283  1832  1877  1922  1967  2012  2057  2102  2147  2192  2237  2282  2327  2372
          2417  2462  2507  2552  2597  2642  2687  2732  2777  2822  2867  2912  2957  3002  3047
          3094  3140  3186  3232  3278  3324  3370  3416  3462  3507  3552  3597  3642  3687  3732
          3777  3822  4127  4170  4252  4295  4544  4606  4722  5213  5261  5453  5472  5690  6867
          6931  6951  7009  7029  7087  7107  7165  7185  7257  7275  7292  7362  7380  7397  7467
          7485  7502  7572  7590  7607  7677  7695  7712  7775  7797  7857  7879  7939  7961  8021
          8043  8115  8132  8149  8218  8235  8252  8321  8338  8355  8424  8441  8458  8527  8544
          8561  8619  9124  9126  9369  9418  9442  9464  9470  9482  9488  9495  9529  9531  9544
CMPB      5628  9258  9424  9428  9548  9554  9596  9611  9613  9621  9642  9646  9665
DEC       3937  4042  6947  7025  7103  7181  7231  7243  7336  7348  7441  7453  7546  7558  7651
          7663  8089  8101  8192  8204  8295  8307  8398  8410  8501  8513  8657  9295
DECB      9200  9211  9625  9628
EMT        768
HALT       750  9134  9267  9592  9726  9748
INC       1315  3927  4032  4601  4761  4800  4839  4878  4917  4956  4994  4997  5040  5618  5623
          5674  5749  5793  5835  5877  5919  5961  6003  5046  6924  7002  7080  7158  7229  7241
          7334  7346  7439  7451  7544  7556  7649  7661  7757  7768  7793  7839  7850  7875  7921
          7932  7957  8003  8014  8039  8087  8099  8190  8202  8293  8305  8396  8408  8499  8511
          8605  8655  8747  8756  8829  8838  8912  8921  8995  9004  9078  9087  9206  9214  9249
```

```
        9355  9441  9512  9695  9734
INCB    5511  5752  9243  9446  9648
IOT      769
JMP      754   756   757   758   759   760   761  8614  8672
JSR     8667  9255  9261  9505  9601  9620  9627  9634  9684
MOV     1258  1262  1264  1265  1266  1267  1268  1269  1270  1271  1275  1276  1279  1280  1281
        1282  1287  1288  1289  1291  1297  1307  1308  1309  1310  1317  1319  1321  1323  1325
        1327  1329  1331  1333  1335  1337  1338  1339  1340  1363  1364  1370  1371  1380  1397
        1416  1417  1426  1443  1462  1463  1472  1489  1508  1509  1518  1535  1554  1555  1564
        1581  1600  1601  1610  1627  1657  1659  1660  1661  1691  1695  1731  1733  1734  1735
        1762  1764  1765  1766  1794  1796  1797  1798  1827  1830  1831  1847  1872  1875  1876
        1892  1917  1920  1921  1937  1962  1965  1966  1982  2007  2010  2011  2027  2052  2055
        2056  2072  2097  2100  2101  2117  2142  2145  2146  2162  2187  2190  2191  2207  2232
        2235  2236  2252  2277  2280  2281  2297  2322  2325  2326  2342  2367  2370  2371  2387
        2412  2415  2416  2432  2457  2460  2461  2477  2502  2505  2506  2522  2547  2550  2551
        2567  2592  2595  2596  2612  2637  2640  2641  2657  2682  2685  2686  2702  2727  2730
        2731  2747  2772  2775  2776  2792  2817  2820  2821  2837  2862  2865  2866  2882  2907
        2910  2911  2927  2952  2955  2956  2972  2997  3000  3001  3017  3042  3045  3046  3062
        3089  3092  3093  3109  3135  3138  3139  3155  3181  3184  3185  3201  3227  3230  3231
        3247  3273  3276  3277  3293  3319  3322  3323  3339  3365  3368  3369  3385  3411  3414
        3415  3431  3457  3460  3461  3477  3502  3505  3506  3522  3547  3550  3551  3567  3592
        3595  3596  3612  3637  3640  3641  3657  3682  3685  3686  3702  3727  3730  3731  3747
        3772  3775  3776  3792  3817  3820  3821  3837  3874  3884  3923  3939  3940  3979  3987
        3989  4029  4044  4045  4078  4086  4115  4118  4123  4125  4158  4161  4166  4168  4202
        4210  4240  4243  4248  4250  4283  4286  4291  4293  4325  4328  4332  4361  4364  4368
        4396  4399  4403  4432  4435  4439  4500  4538  4542  4543  4584  4586  4591  4596  4604
        4651  4652  4700  4701  4755  4759  4794  4798  4833  4837  4872  4876  4911  4915  4950
        4954  4990  5044  5078  5079  5115  5116  5163  5205  5207  5210  5212  5253  5255  5258
        5260  5287  5290  5296  5324  5325  5355  5356  5389  5390  5435  5436  5437  5451  5452
        5471  5504  5507  5509  5555  5602  5604  5610  5611  5616  5626  5671  5672  5717  5721
        5726  5744  5785  5790  5827  5832  5869  5874  5911  5916  5953  5958  5995  6000  6042
        6044  6047  6137  6139  6157  6158  6161  6187  6190  6234  6235  6242  6243  6250  6251
        6296  6298  6299  6305  6306  6353  6359  6395  6396  6403  6435  6436  6437  6443  6474
        6475  6531  6533  6551  6552  6554  6581  6584  6627  6628  6635  6636  6643  6644  6691
        6693  6694  6700  6701  6747  6748  6749  6752  6779  6783  6791  6826  6827  6857  6865
        6866  6907  6912  6915  6927  6929  6944  6950  6985  6990  6993  7005  7007  7022  7028
        7063  7068  7071  7083  7085  7100  7106  7141  7146  7149  7161  7163  7178  7184  7221
        7226  7227  7235  7238  7239  7247  7249  7273  7326  7331  7332  7340  7343  7344  7352
        7354  7378  7431  7436  7437  7445  7448  7449  7457  7459  7483  7536  7541  7542  7550
        7553  7554  7562  7564  7588  7641  7646  7647  7655  7658  7659  7667  7669  7693  7750
        7755  7758  7771  7773  7790  7796  7832  7837  7840  7853  7855  7872  7878  7914  7919
        7922  7935  7937  7954  7960  7996  8001  8004  8017  8019  8036  8042  8079  8084  8085
        8093  8096  8097  8105  8107  8130  8182  8187  8188  8196  8199  8200  8208  8210  8233
        8285  8290  8291  8299  8302  8303  8311  8313  8336  8388  8393  8394  8402  8405  8406
        8414  8416  8439  8491  8496  8497  8505  8508  8509  8517  8519  8542  8589  8590  8599
        8610  8612  8618  8626  8633  8634  8635  8636  8660  8664  8712  8794  8802  8803  8877
        8956  8964  8966  9039  9047  9049  9102  9109  9119  9121  9175  9183  9184  9185  9191
        9198  9216  9217  9218  9219  9220  9245  9250  9270  9273  9286  9291  9300  9305  9310
        9312  9316  9338  9339  9340  9341  9342  9343  9344  9349  9352  9372  9378  9379  9380
        9381  9382  9384  9385  9413  9414  9416  9419  9432  9444  9445  9448  9449  9452  9453
        9475  9492  9499  9501  9523  9524  9542  9543  9558  9559  9560  9561  9594  9595  9600
        9608  9623  9661  9662  9669  9673  9678  9679  9681  9683  9693  9699  9700  9715  9716
        9717  9718  9719  9720  9721  9722  9723  9724  9725  9731  9732  9736  9737  9738  9739
        9740  9741  9742  9743  9744  9763  9764  9768
MOVB    1274  1365  3877  3882  3931  3982  4036  9176  9177  9180  9181  9182  9186  9189  9190
        9209  9252  9260  9347  9350  9364  9367  9376  9447  9451  9468  9527  9547  9552  9605
```

```
                9633  9641  9656  9657  9659  9766
NEG             7254  7359  7464  7569  7674  8112  8215  8318  8421  8524  9187  9346
NOP             1362  6168  6169  6193  6257  6258  6311  6362  6363  6406  6407  6446  6447  6561  6562
                6587  6649  6650  6706  6755  6758  6759  6794  6795  8652  8668  8669  8670
RESET           4330  4366  4401  4437  5294  5723  6312  6707  8666
ROL             9193  9195  9196  9197  9199
RTI             1290  1341  6162  6191  6252  6307  6360  6404  6444  6555  6585  6645  6702  6753  6792
                9128  9221  9275  9386  9454  9504  9534  9562  9610  9747
RTS             9314  9650  9701  9769
SUB             4626  5648  6249  6642  7250  7355  7460  7565  7670  8108  8211  8314  8417  8520  9103
                9120  9251  9353  9676
SWAB            6244  6300  6637  6695
TRAP            9771  9781  9782  9783  9784  9785  9786  9787  9788
TST             1286  1299  1374  1420  1466  1512  1558  1604  4471  4621  4763  4802  4841  4880  4919
                4958  4999  5035  5047  5121  5161  5203  5251  5327  5358  5394  5515  5560  5756  5796
                5838  5880  5922  5964  6006  6053  6133  6135  6230  6232  6527  6529  6623  6625  6918
                6996  7074  7152  7761  7843  7925  8007  8592  8811  8895  8977  9060  9204  9265  9271
                9318  9358  9368  9415  9439  9490  9497  9507  9607  9615  9637  9671  9689  9691  9765
TSTB            3887  3942  3992  4047  5517  5643  5677  5799  5841  5883  5925  5967  6009  8585  9360
                9374  9426  9466  9525  9590  9639  9663  9674  9687
.ASCII           966   967  9906
.ASCIZ           965   968  1305  6144  6239  6538  6632  8598  8604  8609  8625  8631  8675  8754  8761
                8836  8843  8919  8926  9002  9009  9085  9092  9107  9115  9322  9566  9567  9568  9570
                9751  9789  9794  9798  9802  9806  9810  9814  9818  9822  9826  9832  9838  9842  9847
                9852  9858  9864  9870  9876  9882  9888  9894  9900  9915  9923  9926  9929  9935  9941
.BLKB           9565
.BLKW           9391
.BYTE            933   934   939   940   955   956   957   958   985   986   996   997  1004  1005  1007
                1008  1010  1011  1013  1014  1015  1016  8674  9222  9223  9224  9225  9262  9263  9563
                9564  9702  9703  9704
.ENABL           719
.END            9979
.ENDC            725   739   741   742   743   753   768   860   874   891   895   897   903   905   912
                 927   931   933   959   962   963   964   965   966   970   974   996  1004  1007  1010
                1013  1014  1015  1016  1019  1020  1021  1022  1023  1024  1025  1026  1027  1028  1029
                1030  1031  1032  1033  1034  1035  1036  1037  1038  1042  1061  1255  1262  1263  1266
                1268  1270  1272  1273  1275  1277  1299  1305  1351  1352  1361  1362  1363  1364  1365
                1367  1383  1386  1394  1397  1401  1402  1411  1412  1413  1429  1432  1440  1443  1447
                1448  1457  1458  1459  1475  1478  1486  1489  1493  1494  1503  1504  1505  1521  1524
                1532  1535  1539  1540  1549  1550  1551  1567  1570  1578  1581  1585  1586  1595  1596
                1597  1613  1616  1624  1627  1632  1633  1655  1656  1657  1658  1666  1669  1672  1675
                1678  1679  1689  1690  1691  1692  1699  1702  1705  1708  1712  1713  1729  1730  1731
                1732  1738  1741  1745  1748  1752  1753  1760  1761  1762  1763  1770  1773  1777  1780
                1785  1786  1792  1793  1794  1795  1802  1805  1809  1812  1817  1818  1825  1826  1827
                1828  1835  1838  1841  1844  1850  1853  1856  1859  1862  1863  1870  1871  1872  1873
                1880  1883  1886  1889  1895  1898  1901  1904  1907  1908  1915  1916  1917  1918  1925
                1928  1931  1934  1940  1943  1946  1949  1952  1953  1960  1961  1962  1963  1970  1973
                1976  1979  1985  1988  1991  1994  1997  1998  2005  2006  2007  2008  2015  2018  2021
                2024  2030  2033  2036  2039  2042  2043  2050  2051  2052  2053  2060  2063  2066  2069
                2075  2078  2081  2084  2087  2088  2095  2096  2097  2098  2105  2108  2111  2114  2120
                2123  2126  2129  2132  2133  2140  2141  2142  2143  2150  2153  2156  2159  2165  2168
                2171  2174  2177  2178  2185  2186  2187  2188  2195  2198  2201  2204  2210  2213  2216
                2219  2222  2223  2230  2231  2232  2233  2240  2243  2246  2249  2255  2258  2261  2264
                2267  2268  2275  2276  2277  2278  2285  2288  2291  2294  2300  2303  2306  2309  2312
                2313  2320  2321  2322  2323  2330  2333  2336  2339  2345  2348  2351  2354  2357  2358
                2365  2366  2367  2368  2375  2378  2381  2384  2390  2393  2396  2399  2402  2403  2410
```

2411	2412	2413	2420	2423	2426	2429	2435	2438	2441	2444	2447	2448	2455	2456
2457	2458	2465	2468	2471	2474	2480	2483	2486	2489	2492	2493	2500	2501	2502
2503	2510	2513	2516	2519	2525	2528	2531	2534	2537	2538	2545	2546	2547	2548
2555	2558	2561	2564	2570	2573	2576	2579	2582	2583	2590	2591	2592	2593	2600
2603	2606	2609	2615	2618	2621	2624	2627	2628	2635	2636	2637	2638	2645	2648
2651	2654	2660	2663	2666	2669	2672	2673	2680	2681	2682	2693	2690	2693	2696
2699	2705	2708	2711	2714	2717	2718	2725	2726	2727	2728	2735	2738	2741	2744
2750	2753	2756	2759	2762	2763	2770	2771	2772	2773	2780	2783	2786	2789	2795
2798	2801	2804	2807	2808	2815	2816	2817	2818	2825	2828	2831	2834	2840	2843
2846	2849	2852	2853	2860	2861	2862	2863	2870	2873	2876	2879	2885	2888	2891
2894	2897	2898	2905	2906	2907	2908	2915	2918	2921	2924	2930	2933	2936	2939
2942	2943	2950	2951	2952	2953	2960	2963	2966	2969	2975	2978	2981	2984	2987
2988	2995	2996	2997	2998	3005	3008	3011	3014	3020	3023	3026	3029	3032	3033
3040	3041	3042	3043	3050	3053	3056	3059	3065	3068	3071	3074	3079	3080	3087
3088	3089	3090	3097	3100	3103	3106	3112	3115	3118	3121	3125	3126	3133	3134
3135	3136	3143	3146	3149	3152	3158	3161	3164	3167	3171	3172	3179	3180	3181
3182	3189	3192	3195	3198	3204	3207	3210	3213	3217	3218	3225	3226	3227	3228
3235	3238	3241	3244	3250	3253	3256	3259	3263	3264	3271	3272	3273	3274	3281
3284	3287	3290	3296	3299	3302	3305	3309	3310	3317	3318	3319	3320	3327	3330
3333	3336	3342	3345	3348	3351	3355	3356	3363	3364	3365	3366	3373	3376	3379
3382	3388	3391	3394	3397	3401	3402	3409	3410	3411	3412	3419	3422	3425	3428
3434	3437	3440	3443	3447	3448	3455	3456	3457	3458	3465	3468	3471	3474	3480
3483	3486	3489	3492	3493	3500	3501	3502	3503	3510	3513	3516	3519	3525	3528
3531	3534	3537	3538	3545	3546	3547	3548	3555	3558	3561	3564	3570	3573	3576
3579	3582	3583	3590	3591	3592	3593	3600	3603	3606	3609	3615	3618	3621	3624
3627	3628	3635	3636	3637	3638	3645	3648	3651	3654	3660	3663	3666	3669	3672
3673	3680	3681	3682	3683	3690	3693	3696	3699	3705	3708	3711	3714	3717	3718
3725	3726	3727	3728	3735	3738	3741	3744	3750	3753	3756	3759	3762	3763	3770
3771	3772	3773	3780	3783	3786	3789	3795	3798	3801	3804	3807	3808	3815	3816
3817	3818	3825	3828	3831	3834	3840	3843	3846	3849	3857	3858	3872	3873	3874
3875	3892	3895	3900	3903	3906	3907	3921	3922	3923	3924	3946	3949	3954	3957
3962	3963	3977	3978	3979	3980	3997	4000	4005	4008	4012	4013	4027	4028	4029
4030	4051	4054	4059	4062	4066	4067	4076	4077	4078	4079	4090	4093	4097	4100
4103	4104	4113	4114	4115	4116	4132	4135	4140	4143	4146	4147	4156	4157	4158
4159	4175	4178	4183	4186	4190	4191	4200	4201	4202	4203	4214	4217	4221	4224
4228	4229	4238	4239	4240	4241	4257	4260	4265	4268	4271	4272	4281	4282	4283
4284	4300	4303	4308	4311	4315	4316	4323	4324	4325	4326	4337	4340	4344	4347
4350	4351	4359	4360	4361	4362	4373	4376	4380	4383	4386	4387	4394	4395	4396
4397	4408	4411	4415	4418	4421	4422	4430	4431	4432	4433	4444	4447	4451	4454
4458	4459	4465	4466	4467	4475	4478	4482	4485	4489	4490	4497	4498	4499	4506
4509	4514	4517	4521	4522	4533	4534	4535	4548	4551	4558	4561	4565	4566	4582
4583	4584	4585	4610	4613	4617	4620	4637	4638	4647	4648	4649	4658	4661	4664
4667	4673	4676	4681	4684	4686	4687	4696	4697	4698	4711	4714	4717	4720	4726
4729	4737	4740	4745	4746	4753	4754	4755	4756	4771	4774	4777	4780	4784	4785
4792	4793	4794	4795	4810	4813	4816	4819	4823	4824	4831	4832	4833	4834	4849
4852	4855	4858	4862	4863	4870	4871	4872	4873	4888	4891	4894	4897	4901	4902
4909	4910	4911	4912	4927	4930	4933	4936	4940	4941	4948	4949	4950	4951	4966
4969	4972	4975	4979	4980	4988	4989	4990	4991	5007	5010	5015	5018	5022	5023
5032	5033	5034	5051	5054	5058	5061	5065	5066	5074	5075	5076	5086	5089	5092
5095	5099	5100	5111	5112	5113	5124	5127	5139	5142	5146	5147	5158	5159	5160
5171	5174	5176	5179	5183	5184	5200	5201	5202	5216	5219	5224	5227	5231	5232
5248	5249	5250	5264	5267	5272	5275	5280	5281	5285	5286	5287	5288	5299	5302
5305	5308	5312	5313	5320	5321	5322	5330	5333	5337	5340	5343	5344	5351	5352
5353	5361	5364	5368	5371	5375	5376	5386	5387	5388	5397	5400	5403	5406	5410
5411	5430	5431	5432	5456	5459	5466	5469	5477	5480	5486	5489	5492	5493	5502
5503	5504	5505	5520	5523	5526	5529	5536	5537	5552	5553	5554	5563	5566	5579

```
                5582    5586    5587    5600    5601    5602    5603    5631    5634    5638    5641    5655    5656    5666    5667
                5668    5680    5683    5686    5689    5694    5697    5700    5703    5707    5708    5715    5716    5717    5718
                5730    5733    5736    5739    5741    5742    5743    5744    5745    5759    5762    5766    5769    5775    5776
                5783    5784    5785    5786    5804    5807    5810    5813    5817    5818    5825    5826    5827    5828    5846
                5849    5852    5855    5859    5860    5867    5868    5869    5870    5888    5891    5894    5897    5901    5902
                5909    5910    5911    5912    5930    5933    5936    5939    5943    5944    5951    5952    5953    5954    5972
                5975    5978    5981    5985    5986    5993    5994    5995    5996    6014    6017    6020    6023    6027    6028
                6036    6037    6038    6056    6059    6070    6073    6085    6086    6128    6129    6130    6144    6171    6174
                6178    6181    6199    6202    6206    6209    6213    6214    6226    6227    6228    6239    6260    6263    6268
                6271    6281    6282    6294    6295    6296    6297    6321    6324    6330    6333    6338    6339    6348    6349
                6350    6382    6383    6390    6391    6392    6422    6423    6430    6431    6432    6449    6452    6463    6466
                6479    6480    6522    6523    6524    6538    6565    6568    6572    6575    6593    6596    6600    6603    6607
                6608    6619    6620    6621    6632    6653    6656    6661    6664    6676    6677    6689    6690    6691    6692
                6716    6719    6725    6728    6732    6733    6742    6743    6744    6761    6764    6772    6775    6801    6804
                6822    6825    6837    6838    6854    6855    6856    6871    6874    6881    6884    6891    6892    6905    6906
                6907    6908    6934    6937    6940    6943    6955    6958    6962    6965    6969    6970    6983    6984    6985
                6986    7012    7015    7018    7021    7033    7036    7040    7043    7047    7048    7061    7062    7063    7064
                7090    7093    7096    7099    7111    7114    7118    7121    7125    7126    7139    7140    7141    7142    7168
                7171    7174    7177    7189    7192    7196    7199    7208    7209    7219    7220    7221    7222    7262    7265
                7268    7271    7280    7283    7287    7290    7296    7299    7303    7306    7313    7314    7324    7325    7326
                7327    7367    7370    7373    7376    7385    7388    7392    7395    7401    7404    7408    7411    7418    7419
                7429    7430    7431    7432    7472    7475    7478    7481    7490    7493    7497    7500    7506    7509    7513
                7516    7523    7524    7534    7535    7536    7537    7577    7580    7583    7586    7595    7598    7602    7605
                7611    7614    7618    7621    7628    7629    7639    7640    7641    7642    7682    7685    7688    7691    7700
                7703    7707    7710    7716    7719    7723    7726    7733    7734    7748    7749    7750    7751    7778    7781
                7785    7788    7801    7804    7808    7811    7815    7816    7830    7831    7832    7833    7860    7863    7867
                7870    7883    7886    7890    7893    7897    7898    7912    7913    7914    7915    7942    7945    7949    7952
                7965    7968    7972    7975    7979    7980    7994    7995    7996    7997    8024    8027    8031    8034    8047
                8050    8054    8057    8066    8067    8077    8078    8079    8080    8119    8122    8125    8128    8137    8140
                8144    8147    8153    8156    8160    8163    8169    8170    8180    8181    8182    8183    8222    8225    8228
                8231    8240    8243    8247    8250    8256    8259    8263    8266    8272    8273    8283    8284    8285    8286
                8325    8328    8331    8334    8343    8346    8350    8353    8359    8362    8366    8369    8375    8376    8386
                8387    8388    8389    8428    8431    8434    8437    8446    8449    8453    8456    8462    8465    8469    8472
                8478    8479    8489    8490    8491    8492    8531    8534    8537    8540    8549    8552    8556    8559    8565
                8568    8572    8575    8598    8604    8609    8625    8631    8644    8645    8646    8648    8650    8653    8659
                8662    8663    8664    8666    8672    8674    8677    8732    8735    8739    8742    8754    8761    8815    8818
                8821    8824    8836    8843    8898    8901    8904    8907    8919    8926    8980    8983    8987    8990    9002
                9009    9063    9066    9070    9073    9085    9092    9099    9107    9115    9131    9134    9140    9143    9153
                9231    9234    9243    9250    9255    9256    9257    9265    9275    9276    9280    9295    9324    9328    9396
                9399    9404    9408    9410    9421    9424    9425    9426    9428    9430    9437    9441    9446    9448    9452
                9455    9456    9460    9516    9535    9536    9543    9545    9549    9550    9566    9576    9605    9656    9657
                9660    9687    9702    9714    9723    9724    9730    9736    9737    9747    9754    9758    9764    9767    9780
                9781    9782    9783    9784    9785    9786    9787    9788

.EQUIV           768     769     777     792     793     822     823     824     825     826     827     828     829     830     831
                 850     851     852     853     854     855     856     857     858     859

.EVEN            974    1017    1305    6144    6239    6538    6632    8598    8604    8609    8625    8631    8754    8761    8836
                8843    8919    8926    9002    9009    9085    9092    9107    9115    9323    9705    9753    9946

.IDENT           719
.IF              721     739     740     741     742     743     753     766     832     860     890     893     895     902     904
                 911     926     930     932     959     962     963     964     965     969     970     973     996    1004    1007
                1010    1013    1014    1015    1016    1019    1020    1021    1022    1023    1024    1025    1026    1027    1028
                1029    1030    1031    1032    1033    1034    1035    1036    1037    1038    1042    1255    1257    1262    1264
                1266    1268    1270    1272    1273    1275    1293    1304    1349    1350    1352    1361    1363    1364    1365
                1383    1386    1394    1397    1400    1402    1411    1413    1429    1432    1440    1443    1446    1448    1457
                1459    1475    1478    1486    1489    1492    1494    1503    1505    1521    1524    1532    1535    1538    1540
                1549    1551    1567    1570    1578    1581    1584    1586    1595    1597    1613    1616    1624    1627    1631
```

1633	1655	1657	1658	1666	1669	1672	1675	1677	1679	1689	1691	1692	1699	1702
1705	1708	1711	1713	1729	1731	1732	1738	1741	1745	1748	1751	1753	1760	1762
1763	1770	1773	1777	1780	1784	1786	1792	1794	1795	1802	1805	1809	1812	1816
1818	1825	1827	1828	1835	1838	1841	1844	1850	1853	1856	1859	1861	1863	1870
1872	1873	1880	1883	1886	1889	1895	1898	1901	1904	1906	1908	1915	1917	1918
1925	1928	1931	1934	1940	1943	1946	1949	1951	1953	1960	1962	1963	1970	1973
1976	1979	1985	1988	1991	1994	1996	1998	2005	2007	2008	2015	2018	2021	2024
2030	2033	2036	2039	2041	2043	2050	2052	2053	2060	2063	2066	2069	2075	2078
2081	2084	2086	2088	2095	2097	2098	2105	2108	2111	2114	2120	2123	2126	2129
2131	2133	2140	2142	2143	2150	2153	2156	2159	2165	2168	2171	2174	2176	2178
2185	2187	2188	2195	2198	2201	2204	2210	2213	2216	2219	2221	2223	2230	2232
2233	2240	2243	2246	2249	2255	2258	2261	2264	2266	2268	2275	2277	2278	2285
2288	2291	2294	2300	2303	2306	2309	2311	2313	2320	2322	2323	2330	2333	2336
2339	2345	2348	2351	2354	2356	2358	2365	2367	2368	2375	2378	2381	2384	2390
2393	2396	2399	2401	2403	2410	2412	2413	2420	2423	2426	2429	2435	2438	2441
2444	2446	2448	2455	2457	2458	2465	2468	2471	2474	2480	2483	2486	2489	2491
2493	2500	2502	2503	2510	2513	2516	2519	2525	2528	2531	2534	2536	2538	2545
2547	2548	2555	2558	2561	2564	2570	2573	2576	2579	2581	2583	2590	2592	2593
2600	2603	2606	2609	2615	2618	2621	2624	2626	2628	2635	2637	2638	2645	2648
2651	2654	2660	2663	2666	2669	2671	2673	2680	2682	2683	2690	2693	2696	2699
2705	2708	2711	2714	2716	2718	2725	2727	2728	2735	2738	2741	2744	2750	2753
2756	2759	2761	2763	2770	2772	2773	2780	2783	2786	2789	2795	2798	2801	2804
2806	2808	2815	2817	2818	2825	2828	2831	2834	2840	2843	2846	2849	2851	2853
2860	2862	2863	2870	2873	2876	2879	2885	2888	2891	2894	2896	2898	2905	2907
2908	2915	2918	2921	2924	2930	2933	2936	2939	2941	2913	2950	2952	2953	2960
2963	2966	2969	2975	2978	2981	2984	2986	2988	2995	2997	2998	3005	3008	3011
3014	3020	3023	3026	3029	3031	3033	3040	3042	3043	3050	3053	3056	3059	3065
3068	3071	3074	3078	3080	3087	3089	3090	3097	3100	3103	3106	3112	3115	3118
3121	3124	3126	3133	3135	3136	3143	3146	3149	3152	3158	3161	3164	3167	3170
3172	3179	3181	3182	3189	3192	3195	3198	3204	3207	3210	3213	3216	3218	3225
3227	3228	3235	3238	3241	3244	3250	3253	3256	3259	3262	3264	3271	3273	3274
3281	3284	3287	3290	3296	3299	3302	3305	3308	3310	3317	3319	3320	3327	3330
3333	3336	3342	3345	3348	3351	3354	3356	3363	3365	3366	3373	3376	3379	3382
3388	3391	3394	3397	3400	3402	3409	3411	3412	3419	3422	3425	3428	3434	3437
3440	3443	3446	3448	3455	3457	3458	3465	3468	3471	3474	3480	3483	3486	3489
3491	3493	3500	3502	3503	3510	3513	3516	3519	3525	3528	3531	3534	3536	3538
3545	3547	3548	3555	3558	3561	3564	3570	3573	3576	3579	3581	3583	3590	3592
3593	3600	3603	3606	3609	3615	3618	3621	3624	3626	3628	3635	3637	3638	3645
3648	3651	3654	3660	3663	3666	3669	3671	3673	3680	3682	3683	3690	3693	3696
3699	3705	3708	3711	3714	3716	3718	3725	3727	3728	3735	3738	3741	3744	3750
3753	3756	3759	3761	3763	3770	3772	3773	3780	3783	3786	3789	3795	3798	3801
3804	3806	3808	3815	3817	3818	3825	3828	3831	3834	3840	3843	3846	3849	3856
3858	3872	3874	3875	3892	3895	3900	3903	3905	3907	3921	3923	3924	3946	3949
3954	3957	3961	3963	3977	3979	3980	3997	4000	4005	4008	4011	4013	4027	4029
4030	4051	4054	4059	4062	4065	4067	4076	4078	4079	4090	4093	4097	4100	4102
4104	4113	4115	4116	4132	4135	4140	4143	4145	4147	4156	4158	4159	4175	4178
4183	4186	4189	4191	4200	4202	4203	4214	4217	4221	4224	4227	4229	4238	4240
4241	4257	4260	4265	4268	4270	4272	4281	4283	4284	4300	4303	4308	4311	4314
4316	4323	4325	4326	4337	4340	4344	4347	4349	4351	4359	4361	4362	4373	4376
4380	4383	4385	4387	4394	4396	4397	4408	4411	4415	4418	4420	4422	4430	4432
4433	4444	4447	4451	4454	4457	4459	4465	4467	4475	4478	4482	4485	4488	4490
4497	4499	4506	4509	4514	4517	4520	4522	4533	4535	4548	4551	4558	4561	4564
4566	4582	4584	4585	4610	4613	4617	4620	4636	4638	4647	4649	4658	4661	4664
4667	4673	4676	4681	4684	4685	4687	4696	4698	4711	4714	4717	4720	4726	4729
4737	4740	4744	4746	4753	4755	4756	4771	4774	4777	4780	4783	4785	4792	4794
4795	4810	4813	4816	4819	4822	4824	4831	4833	4834	4849	4852	4855	4858	4861

4863	4870	4872	4873	4888	4891	4894	4897	4900	4902	4909	4911	4912	4927	4930
4933	4936	4939	4941	4948	4950	4951	4966	4969	4972	4975	4978	4980	4988	4990
4991	5007	5010	5015	5018	5021	5023	5032	5034	5051	5054	5058	5061	5064	5066
5074	5076	5086	5089	5092	5095	5098	5100	5111	5113	5124	5127	5139	5142	5145
5147	5158	5160	5171	5174	5176	5179	5182	5184	5200	5202	5216	5219	5224	5227
5230	5232	5248	5250	5264	5267	5272	5275	5279	5281	5285	5287	5288	5299	5302
5305	5308	5311	5313	5320	5322	5330	5333	5337	5340	5342	5344	5351	5353	5361
5364	5368	5371	5374	5376	5386	5388	5397	5400	5403	5406	5409	5411	5430	5432
5456	5459	5466	5469	5477	5480	5486	5489	5491	5493	5502	5504	5505	5520	5523
5526	5529	5535	5537	5552	5554	5563	5566	5579	5582	5585	5587	5600	5602	5603
5631	5634	5638	5641	5654	5656	5666	5668	5680	5683	5686	5689	5694	5697	5700
5703	5706	5708	5715	5717	5718	5730	5733	5736	5739	5740	5742	5744	5745	5759
5762	5766	5769	5774	5776	5783	5785	5786	5804	5807	5810	5813	5816	5818	5825
5827	5828	5846	5849	5852	5855	5858	5860	5867	5869	5870	5888	5891	5894	5897
5900	5902	5909	5911	5912	5930	5933	5936	5939	5942	5944	5951	5953	5954	5972
5975	5978	5981	5984	5986	5993	5995	5996	6014	6017	6020	6023	6026	6028	6036
6038	6056	6059	6070	6073	6084	6086	6128	6130	6143	6171	6174	6178	6181	6199
6202	6206	6209	6212	6214	6226	6228	6238	6260	6263	6268	6271	6280	6282	6294
6296	6297	6321	6324	6330	6333	6337	6339	6348	6350	6381	6383	6390	6392	6421
6423	6430	6432	6449	6452	6463	6466	6478	6480	6522	6524	6537	6565	6568	6572
6575	6593	6596	6600	6603	6606	6608	6619	6621	6631	6653	6656	6661	6664	6675
6677	6689	6691	6692	6716	6719	6725	6728	6731	6733	6742	6744	6761	6764	6772
6775	6801	6804	6822	6825	6836	6838	6854	6856	6871	6874	6881	6884	6890	6892
6905	6907	6908	6934	6937	6940	6943	6955	6958	6962	6965	6968	6970	6983	6985
6986	7012	7015	7018	7021	7033	7036	7040	7043	7046	7048	7061	7063	7064	7090
7093	7096	7099	7111	7114	7118	7121	7124	7126	7139	7141	7142	7168	7171	7174
7177	7189	7192	7196	7199	7207	7209	7219	7221	7222	7262	7265	7268	7271	7280
7283	7287	7290	7296	7299	7303	7306	7312	7314	7324	7326	7327	7367	7370	7373
7376	7385	7388	7392	7395	7401	7404	7408	7411	7417	7419	7429	7431	7432	7472
7475	7478	7481	7490	7493	7497	7500	7506	7509	7513	7516	7522	7524	7534	7536
7537	7577	7580	7583	7586	7595	7598	7602	7605	7611	7614	7618	7621	7627	7629
7639	7641	7642	7682	7685	7688	7691	7700	7703	7707	7710	7716	7719	7723	7726
7732	7734	7748	7750	7751	7778	7781	7785	7788	7801	7804	7808	7811	7814	7816
7830	7832	7833	7860	7863	7867	7870	7883	7886	7890	7893	7896	7898	7912	7914
7915	7942	7945	7949	7952	7965	7968	7972	7975	7978	7980	7994	7996	7997	8024
8027	8031	8034	8047	8050	8054	8057	8065	8067	8077	8079	8080	8119	8122	8125
8128	8137	8140	8144	8147	8153	8156	8160	8163	8168	8170	8180	8182	8183	8222
8225	8228	8231	8240	8243	8247	8250	8256	8259	8263	8266	8271	8273	8283	8285
8286	8325	8328	8331	8334	8343	8346	8350	8353	8359	8362	8366	8369	8374	8376
8386	8388	8389	8428	8431	8434	8437	8446	8449	8453	8456	8462	8465	8469	8472
8477	8479	8489	8491	8492	8531	8534	8537	8540	8549	8552	8556	8559	8565	8568
8572	8575	8597	8603	8608	8624	8630	8643	8644	8645	8646	8647	8648	8652	8658
8661	8663	8664	8666	8672	8674	8675	8677	8732	8735	8739	8742	8753	8760	8815
8818	8821	8824	8835	8842	8898	8901	8904	8907	8918	8925	8980	8983	8987	8990
9001	9008	9063	9066	9070	9073	9084	9091	9099	9106	9114	9131	9134	9140	9143
9152	9230	9233	9243	9246	9253	9255	9256	9258	9265	9268	9275	9276	9279	9294
9310	9327	9395	9398	9403	9408	9420	9422	9423	9424	9426	9427	9428	9437	9439
9447	9449	9454	9455	9456	9459	9460	9515	9535	9543	9544	9548	9549	9565	9566
9575	9596	9655	9657	9660	9687	9702	9713	9723	9724	9729	9736	9737	9745	9747
9751	9757	9763	9767	9771	9781	9782	9783	9784	9785	9786	9787	9788		
.IFF 739	741	742	743	766	891	895	897	903	905	912	927	930	933	959
970	974	1262	1350	1351	1352	1362	1363	1367	1400	1401	1402	1412	1413	1446
1447	1448	1458	1459	1492	1493	1494	1504	1505	1538	1539	1540	1550	1551	1584
1585	1586	1596	1597	1632	1633	1656	1657	1678	1679	1690	1691	1712	1713	1730
1731	1752	1753	1761	1762	1785	1786	1793	1794	1817	1818	1826	1827	1862	1863
1871	1872	1907	1908	1916	1917	1952	1953	1961	1962	1997	1998	2006	2007	2042

	2043	2051	2052	2087	2088	2096	2097	2132	2133	2141	2142	2177	2178	2186	2187
	2222	2223	2231	2232	2267	2268	2276	2277	2312	2313	2321	2322	2357	2358	2366
	2367	2402	2403	2411	2412	2447	2448	2456	2457	2492	2493	2501	2502	2537	2538
	2546	2547	2582	2583	2591	2592	2627	2628	2636	2637	2672	2673	2681	2682	2717
	2718	2726	2727	2762	2763	2771	2772	2807	2808	2816	2817	2852	2853	2861	2862
	2897	2898	2906	2907	2942	2943	2951	2952	2987	2988	2996	2997	3032	3033	3041
	3042	3079	3080	3088	3089	3125	3126	3134	3135	3171	3172	3180	3181	3217	3218
	3226	3227	3263	3264	3272	3273	3309	3310	3318	3319	3355	3356	3364	3365	3401
	3402	3410	3411	3447	3448	3456	3457	3492	3493	3501	3502	3537	3538	3546	3547
	3582	3583	3591	3592	3627	3628	3636	3637	3672	3673	3681	3682	3717	3718	3726
	3727	3762	3763	3771	3772	3807	3808	3816	3817	3857	3858	3873	3874	3906	3907
	3922	3923	3962	3963	3978	3979	4012	4013	4028	4029	4066	4067	4077	4078	4103
	4104	4114	4115	4146	4147	4157	4158	4190	4191	4201	4202	4228	4229	4239	4240
	4271	4272	4282	4283	4315	4316	4324	4325	4350	4351	4360	4361	4386	4387	4395
	4396	4421	4422	4431	4432	4458	4459	4466	4467	4489	4490	4498	4499	4521	4522
	4534	4535	4565	4566	4583	4584	4585	4637	4638	4648	4649	4686	4687	4697	4698
	4745	4746	4754	4755	4784	4785	4793	4794	4823	4824	4832	4833	4862	4863	4871
	4872	4901	4902	4910	4911	4940	4941	4949	4950	4979	4980	4989	4990	5022	5023
	5033	5034	5065	5066	5075	5076	5099	5100	5112	5113	5146	5147	5159	5160	5183
	5184	5201	5202	5231	5232	5249	5250	5280	5281	5286	5287	5312	5313	5321	5322
	5343	5344	5352	5353	5375	5376	5387	5388	5410	5411	5431	5432	5492	5493	5503
	5504	5536	5537	5553	5554	5586	5587	5601	5602	5603	5655	5656	5667	5668	5707
	5708	5716	5717	5718	5741	5742	5743	5744	5775	5776	5784	5785	5817	5818	5826
	5827	5859	5860	5868	5869	5901	5902	5910	5911	5943	5944	5952	5953	5995	5986
	5994	5995	6027	6028	6037	6038	6085	6086	6129	6130	6213	6214	6227	6228	6281
	6282	6295	6296	6338	6339	6349	6350	6382	6383	6391	6392	6422	6423	6431	6432
	6479	6480	6523	6524	6607	6608	6620	6621	6676	6677	6690	6691	6732	6733	6743
	6744	6837	6838	6855	6856	6891	6892	6906	6907	6969	6970	6984	6985	7047	7048
	7062	7063	7125	7126	7140	7141	7208	7209	7220	7221	7313	7314	7325	7326	7418
	7419	7430	7431	7523	7524	7535	7536	7628	7629	7640	7641	7733	7734	7749	7750
	7815	7816	7831	7832	7897	7898	7913	7914	7979	7980	7995	7996	8066	8067	8078
	8079	8169	8170	8181	8182	8272	8273	8284	8285	8375	8376	8387	8388	8478	8479
	8490	8491	8644	8647	8652	8659	8662	8674	9153	9231	9233	9246	9275	9276	9280
	9295	9324	9328	9396	9421	9424	9425	9428	9455	9456	9460	9482	9516	9518	9523
	9535	9536	9545	9549	9566	9576	9656	9714	9730	9747	9758	9764			
.IFT	1305	6144	6239	6538	6632	8598	8604	8609	8625	8631	8754	8761	8836	8843	8919
	8926	9002	9009	9085	9092	9107	9115	9256	9436	9460	9482	9518	9523		
.IFTF	1305	6144	6239	6538	6632	8598	8604	8609	8625	8631	8754	8761	8836	8843	8919
	8926	9002	9009	9085	9092	9107	9115	9255	9434	9474	9488	9516	9519		
.IIF	720	725	730	731	736	737	738	739	742	743	750	969	974	1263	1266
	1272	1273	1275	1276	1292	8600	8611	8627	8645	8646	8653	8654	8674	8677	9110
	9122	9234	9235	9236	9237	9238	9243	9268	9275	9276	9292	9317	9399	9400	9401
	9402	9403	9404	9408	9435	9436	9452	9455	9456	9460	9476	9558	9566	9572	9652
	9780	9781	9782	9783	9784	9785	9786	9787	9788						
.IRP	1042	1255	1350	1400	1446	1492	1538	1584	1631	1677	1711	1751	1784	1816	1861
	1906	1951	1996	2041	2086	2131	2176	2221	2266	2311	2356	2401	2446	2491	2536
	2581	2626	2671	2716	2761	2806	2851	2896	2941	2986	3031	3078	3124	3170	3216
	3262	3308	3354	3400	3446	3491	3536	3581	3626	3671	3716	3761	3806	3856	3905
	3961	4011	4065	4102	4145	4189	4227	4270	4314	4349	4385	4420	4457	4488	4520
	4564	4636	4685	4744	4783	4822	4861	4900	4939	4978	5021	5064	5098	5145	5182
	5230	5279	5311	5342	5374	5409	5491	5535	5585	5654	5706	5740	5774	5816	5858
	5900	5942	5984	6026	6084	6212	6280	6337	6381	6421	6478	6606	6675	6731	6836
	6890	6968	7046	7124	7207	7312	7417	7522	7627	7732	7814	7896	7978	8065	8168
	8271	8374	8477	8652	9338	9378	9661	9662	9683	9699	9700	9717	9723	9736	9737
.LIST	719	742	750	874	959	961	962	963	970	974	1255	1277	1305	1350	1363
	1383	1386	1394	1397	1400	1413	1429	1432	1440	1443	1446	1459	1475	1478	1486

1489	1492	1505	1521	1524	1532	1535	1538	1551	1567	1570	1578	1581	1584	1597
1613	1616	1624	1627	1631	1657	1666	1669	1672	1675	1677	1691	1699	1702	1705
1708	1711	1731	1738	1741	1745	1748	1751	1762	1770	1773	1777	1780	1784	1794
1802	1805	1809	1812	1815	1816	1827	1828	1835	1838	1841	1844	1850	1853	1856
1859	1860	1861	1872	1873	1880	1883	1886	1889	1895	1898	1901	1904	1905	1906
1917	1918	1925	1928	1931	1934	1940	1943	1946	1949	1950	1951	1962	1963	1970
1973	1976	1979	1985	1988	1991	1994	1995	1996	2007	2008	2015	2018	2021	2024
2030	2033	2036	2039	2040	2041	2052	2053	2060	2063	2066	2069	2075	2078	2081
2084	2085	2086	2097	2098	2105	2108	2111	2114	2120	2123	2126	2129	2130	2131
2142	2143	2150	2153	2156	2159	2165	2168	2171	2174	2175	2176	2187	2188	2195
2198	2201	2204	2210	2213	2216	2219	2220	2221	2232	2233	2240	2243	2246	2249
2255	2258	2261	2264	2265	2266	2277	2278	2285	2288	2291	2294	2300	2303	2306
2309	2310	2311	2322	2323	2330	2333	2336	2339	2345	2348	2351	2354	2355	2356
2367	2368	2375	2378	2381	2384	2390	2393	2396	2399	2400	2401	2412	2413	2420
2423	2426	2429	2435	2438	2441	2444	2445	2446	2457	2458	2461	2468	2471	2474
2480	2483	2486	2489	2490	2491	2502	2503	2510	2513	2516	2519	2525	2528	2531
2534	2535	2536	2547	2548	2555	2558	2561	2564	2570	2573	2576	2579	2580	2581
2592	2593	2600	2603	2606	2609	2615	2618	2621	2624	2625	2626	2637	2638	2645
2648	2651	2654	2660	2663	2666	2669	2670	2671	2682	2683	2690	2693	2696	2699
2705	2708	2711	2714	2715	2716	2727	2728	2735	2738	2741	2744	2750	2753	2756
2759	2760	2761	2772	2773	2780	2783	2786	2789	2795	2798	2801	2804	2805	2806
2817	2818	2825	2828	2831	2834	2840	2843	2846	2849	2850	2851	2862	2863	2870
2873	2876	2879	2885	2888	2891	2894	2895	2896	2907	2908	2915	2918	2921	2924
2930	2933	2936	2939	2940	2941	2952	2953	2960	2963	2966	2969	2975	2978	2981
2984	2985	2986	2997	2998	3005	3008	3011	3014	3020	3023	3025	3029	3030	3031
3042	3043	3050	3053	3056	3059	3065	3068	3071	3074	3075	3077	3078	3089	3090
3097	3100	3103	3106	3112	3115	3118	3121	3123	3124	3135	3136	3143	3146	3149
3152	3158	3161	3164	3167	3169	3170	3181	3182	3189	3192	3195	3198	3204	3207
3210	3213	3215	3216	3227	3228	3235	3238	3241	3244	3250	3253	3256	3259	3261
3262	3273	3274	3281	3284	3287	3290	3296	3299	3302	3305	3307	3308	3319	3320
3327	3330	3333	3336	3342	3345	3348	3351	3353	3354	3365	3366	3373	3376	3379
3382	3388	3391	3394	3397	3399	3400	3411	3412	3419	3422	3425	3428	3434	3437
3440	3443	3445	3446	3457	3458	3465	3468	3471	3474	3480	3483	3486	3489	3490
3491	3502	3503	3510	3513	3516	3519	3525	3528	3531	3534	3535	3536	3547	3548
3555	3558	3561	3564	3570	3573	3576	3579	3580	3581	3592	3593	3600	3603	3606
3609	3615	3618	3621	3624	3625	3626	3637	3638	3645	3648	3651	3654	3660	3663
3666	3669	3670	3671	3682	3683	3690	3693	3696	3699	3705	3708	3711	3714	3715
3716	3727	3728	3735	3738	3741	3744	3750	3753	3756	3759	3760	3761	3772	3773
3780	3783	3786	3789	3795	3798	3801	3804	3805	3806	3817	3818	3825	3828	3831
3834	3840	3843	3846	3849	3850	3856	3874	3892	3895	3900	3903	3905	3923	3946
3949	3954	3957	3961	3979	3997	4000	4005	4008	4011	4029	4051	4054	4059	4062
4065	4078	4090	4093	4097	4100	4102	4115	4132	4135	4140	4143	4145	4158	4175
4178	4183	4186	4189	4202	4214	4217	4221	4224	4227	4240	4257	4260	4265	4268
4270	4283	4300	4303	4308	4311	4314	4325	4337	4340	4344	4347	4349	4361	4373
4376	4380	4383	4385	4396	4408	4411	4415	4418	4420	4432	4444	4447	4451	4454
4457	4467	4475	4478	4482	4485	4488	4499	4506	4509	4514	4517	4520	4535	4548
4551	4558	4561	4564	4584	4610	4613	4617	4620	4636	4649	4658	4661	4664	4667
4673	4676	4681	4684	4685	4698	4711	4714	4717	4720	4726	4729	4737	4740	4742
4744	4755	4771	4774	4777	4780	4781	4783	4794	4810	4813	4816	4819	4820	4822
4833	4849	4852	4855	4858	4859	4861	4872	4888	4891	4894	4897	4898	4900	4911
4927	4930	4933	4936	4937	4939	4950	4966	4969	4972	4975	4978	4990	5007	5010
5015	5018	5021	5034	5051	5054	5058	5061	5064	5076	5086	5089	5092	5095	5098
5113	5124	5127	5139	5142	5145	5160	5171	5174	5176	5179	5182	5202	5216	5219
5224	5227	5230	5250	5264	5267	5272	5275	5279	5287	5299	5302	5305	5309	5311
5322	5330	5333	5337	5340	5342	5353	5361	5364	5368	5371	5374	5388	5397	5400
5403	5406	5409	5432	5456	5459	5466	5469	5477	5480	5486	5489	5491	5504	5520

```
                5523  5526  5529  5535  5554  5563  5566  5579  5582  5585  5602  5631  5634  5638  5641
                5654  5668  5680  5683  5686  5689  5694  5697  5700  5703  5706  5717  5730  5733  5736
                5739  5740  5744  5759  5762  5766  5769  5772  5774  5785  5804  5807  5810  5813  5814
                5816  5827  5846  5849  5852  5855  5856  5858  5869  5888  5891  5894  5897  5898  5900
                5911  5930  5933  5936  5939  5940  5942  5953  5972  5975  5978  5981  5982  5984  5995
                6014  6017  6020  6023  6026  6038  6056  6059  6070  6073  6084  6130  6131  6144  6171
                6174  6178  6181  6199  6202  6206  6209  6212  6228  6239  6260  6263  6268  6271  6280
                6296  6321  6324  6330  6333  6337  6350  6381  6392  6421  6432  6449  6452  6463  6466
                6478  6524  6525  6538  6565  6568  6572  6575  6593  6596  6600  6603  6606  6621  6632
                6653  6656  6661  6664  6675  6691  6716  6719  6725  6728  6731  6744  6761  6764  6772
                6775  6801  6804  6822  6825  6836  6856  6871  6874  6881  6884  6888  6890  6907  6934
                6937  6940  6943  6955  6958  6962  6965  6966  6968  6985  7012  7015  7018  7021  7033
                7036  7040  7043  7044  7046  7063  7090  7093  7096  7099  7111  7114  7118  7121  7122
                7124  7141  7168  7171  7174  7177  7189  7192  7196  7199  7204  7207  7221  7262  7265
                7268  7271  7280  7283  7287  7290  7296  7299  7303  7306  7309  7312  7326  7367  7370
                7373  7376  7385  7388  7392  7395  7401  7404  7408  7411  7414  7417  7431  7472  7475
                7478  7481  7490  7493  7497  7500  7506  7509  7513  7516  7519  7522  7536  7577  7580
                7583  7586  7595  7598  7602  7605  7611  7614  7618  7621  7624  7627  7641  7682  7685
                7688  7691  7700  7703  7707  7710  7716  7719  7723  7726  7730  7732  7750  7778  7781
                7785  7788  7801  7804  7808  7811  7812  7814  7832  7860  7863  7867  7870  7883  7886
                7890  7893  7894  7896  7914  7942  7945  7949  7952  7965  7968  7972  7975  7976  7978
                7996  8024  8027  8031  8034  8047  8050  8054  8057  8062  8065  8079  8119  8122  8125
                8128  8137  8140  8144  8147  8153  8156  8160  8163  8165  8168  8182  8222  8225  8228
                8231  8240  8243  8247  8250  8256  8259  8263  8266  8268  8271  8285  8325  8328  8331
                8334  8343  8346  8350  8353  8359  8362  8366  8369  8371  8374  8388  8428  8431  8434
                8437  8446  8449  8453  8456  8462  8465  8469  8472  8474  8477  8491  8531  8534  8537
                8540  8549  8552  8556  8559  8565  8568  8572  8575  8598  8604  8609  8625  8631  8653
                8666  8732  8735  8739  8742  8754  8761  8815  8818  8821  8824  8836  8843  8898  8901
                8904  8907  8919  8926  8980  8983  8987  8990  9002  9009  9063  9066  9070  9073  9085
                9092  9100  9107  9115  9131  9134  9140  9143  9275  9403  9535  9771  9780  9781  9782
                9783  9784  9785  9786  9787  9788  9789
.MACRO           743   880   881   882   884   886   923  1077  1293  1347  1349  1400  1446  1492  1538
                1584  1630  1676  1710  1750  1783  1814  1815  3075  3854  3959  4064  4188  4313  4348
                4384  4419  4456  4487  4519  4563  4635  4685  4741  4743  4782  4821  4860  4899  4938
                4977  5020  5063  5097  5144  5181  5229  5278  5310  5341  5373  5408  5490  5534  5584
                5653  5705  5771  5772  5814  5856  5898  5940  5982  6025  6078  6080  6082  6083  6211
                6279  6336  6380  6420  6477  6605  6674  6730  6834  6887  6888  6966  7044  7122  7202
                7205  7310  7415  7520  7625  7729  7730  7812  7894  7976  8060  8063  8166  8269  8372
                8475  8682  8683  9771
.MCALL           719   874   970  1277
.NLIST           719   742   750   874   959   961   962   963   970   974  1255  1277  1305  1350  1363
                1383  1386  1394  1397  1400  1413  1429  1432  1440  1443  1446  1459  1475  1478  1486
                1489  1492  1505  1521  1524  1532  1535  1538  1551  1567  1570  1578  1581  1584  1597
                1613  1616  1624  1627  1631  1657  1666  1669  1672  1675  1677  1691  1699  1702  1705
                1708  1711  1731  1738  1741  1745  1748  1751  1762  1770  1773  1777  1780  1784  1794
                1802  1805  1809  1812  1815  1816  1827  1828  1835  1838  1841  1844  1850  1853  1856
                1859  1860  1861  1872  1873  1880  1883  1886  1889  1895  1898  1901  1904  1905  1906
                1917  1918  1925  1928  1931  1934  1940  1943  1946  1949  1950  1951  1962  1963  1970
                1973  1976  1979  1985  1988  1991  1994  1995  1996  2007  2008  2015  2018  2021  2024
                2030  2033  2036  2039  2040  2041  2052  2053  2060  2063  2066  2069  2075  2078  2081
                2084  2085  2086  2097  2098  2105  2108  2111  2114  2120  2123  2126  2129  2130  2131
                2142  2143  2150  2153  2156  2159  2165  2168  2171  2174  2175  2176  2187  2188  2195
                2198  2201  2204  2210  2213  2216  2219  2220  2221  2232  2233  2240  2243  2246  2249
                2255  2258  2261  2264  2265  2266  2277  2278  2285  2288  2291  2294  2300  2303  2306
                2309  2310  2311  2322  2323  2330  2333  2336  2339  2345  2348  2351  2354  2355  2356
                2367  2368  2375  2378  2381  2384  2390  2393  2396  2399  2400  2401  2412  2413  2420
```

2423	2426	2429	2435	2438	2441	2444	2445	2446	2457	2458	2465	2468	2471	2474
2480	2483	2486	2489	2490	2491	2502	2503	2510	2513	2516	2519	2525	2528	2531
2534	2535	2536	2547	2548	2555	2558	2561	2564	2570	2573	2576	2579	2580	2581
2592	2593	2600	2603	2606	2609	2615	2618	2621	2624	2625	2626	2637	2638	2645
2648	2651	2654	2660	2663	2666	2669	2670	2671	2682	2683	2690	2693	2696	2699
2705	2708	2711	2714	2715	2716	2727	2728	2735	2738	2741	2744	2750	2753	275F
2759	2760	2761	2772	2773	2780	2783	2786	2789	2795	2798	2801	2804	2805	280B
2817	2818	2825	2828	2831	2834	2840	2843	2846	2849	2850	2851	2862	2863	2870
2873	2876	2879	2885	2888	2891	2894	2895	2896	2907	2908	2915	2918	2921	2924
2930	2933	2936	2939	2940	2941	2952	2953	2960	2963	2966	2969	2975	2978	2981
2984	2985	2986	2997	2998	3005	3008	3011	3014	3020	3023	3026	3029	3030	3031
3042	3043	3050	3053	3056	3059	3065	3068	3071	3074	3075	3077	3078	3089	3030
3097	3100	3103	3106	3112	3115	3118	3121	3123	3124	3135	3136	3143	3146	3149
3152	3158	3161	3164	3167	3169	3170	3181	3182	3189	3192	3195	3198	3204	3207
3210	3213	3215	3216	3227	3228	3235	3238	3241	3244	3250	3253	3256	3259	3261
3262	3273	3274	3281	3284	3287	3290	3296	3299	3302	3305	3307	3308	3319	3320
3327	3330	3333	3336	3342	3345	3348	3351	3353	3354	3365	3366	3373	3376	3379
3382	3388	3391	3394	3397	3399	3400	3411	3412	3419	3422	3425	3428	3434	3437
3440	3443	3445	3446	3457	3458	3465	3468	3471	3474	3480	3483	3486	3489	3490
3491	3502	3503	3510	3513	3516	3519	3525	3528	3531	3534	3535	3536	3547	3548
3555	3558	3561	3564	3570	3573	3576	3579	3580	3581	3592	3593	3600	3603	3606
3609	3615	3618	3621	3624	3625	3626	3637	3638	3645	3648	3651	3654	3660	3663
3666	3669	3670	3671	3682	3683	3690	3693	3696	3699	3705	3708	3711	3714	3715
3716	3727	3728	3735	3738	3741	3744	3750	3753	3756	3759	3760	3761	3772	3773
3780	3783	3786	3789	3795	3798	3801	3804	3805	3806	3817	3818	3825	3828	3831
3834	3840	3843	3846	3849	3850	3856	3874	3892	3895	3900	3903	3905	3923	3946
3949	3954	3957	3961	3979	3997	4000	4005	4008	4011	4029	4051	4054	4059	4062
4065	4078	4090	4093	4097	4100	4102	4115	4132	4135	4140	4143	4145	4158	4175
4178	4183	4186	4189	4202	4214	4217	4221	4224	4227	4240	4257	4260	4265	4268
4270	4283	4300	4303	4308	4311	4314	4325	4337	4340	4344	4347	4349	4361	4373
4376	4380	4383	4385	4396	4408	4411	4415	4418	4420	4432	4444	4447	4451	4454
4457	4467	4475	4478	4482	4485	4488	4499	4506	4509	4514	4517	4520	4535	4548
4551	4558	4561	4564	4584	4610	4613	4617	4620	4636	4649	4658	4661	4664	4667
4673	4676	4681	4684	4685	4698	4711	4714	4717	4720	4726	4729	4737	4740	4742
4744	4755	4771	4774	4777	4780	4781	4783	4794	4810	4813	4816	4819	4820	4822
4833	4849	4852	4855	4858	4859	4861	4872	4888	4891	4894	4897	4898	4900	4911
4927	4930	4933	4936	4937	4939	4950	4966	4969	4972	4975	4978	4990	5007	5010
5015	5018	5021	5034	5051	5054	5058	5061	5064	5076	5086	5089	5092	5095	5098
5113	5124	5127	5139	5142	5145	5160	5171	5174	5176	5179	5182	5202	5216	5219
5224	5227	5230	5250	5264	5267	5272	5275	5279	5287	5299	5302	5305	5308	5311
5322	5330	5333	5337	5340	5342	5353	5361	5364	5368	5371	5374	5388	5397	5400
5403	5406	5409	5432	5456	5459	5466	5469	5477	5480	5486	5489	5491	5504	5520
5523	5526	5529	5535	5554	5563	5566	5579	5582	5585	5602	5631	5634	5638	5641
5654	5668	5680	5683	5686	5689	5694	5697	5700	5703	5706	5717	5730	5733	5736
5739	5740	5744	5759	5762	5766	5769	5772	5774	5785	5804	5807	5810	5813	5814
5816	5827	5846	5849	5852	5855	5856	5858	5869	5888	5891	5894	5897	5898	5900
5911	5930	5933	5936	5939	5940	5942	5953	5972	5975	5978	5981	5982	5984	5995
6014	6017	6020	6023	6026	6038	6056	6059	6070	6073	6084	6130	6131	6144	6171
6174	6178	6181	6199	6202	6206	6209	6212	6228	6239	6260	6263	6268	6271	6280
6296	6321	6324	6330	6333	6337	6350	6381	6392	6421	6432	6449	6452	6463	6466
6478	6524	6525	6538	6565	6568	6572	6575	6593	6596	6600	6603	6606	6621	6632
6653	6656	6661	6664	6675	6691	6716	6719	6725	6728	6731	6744	6761	6764	6772
6775	6801	6804	6822	6825	6836	6856	6871	6874	6881	6884	6888	6890	6907	6934
6937	6940	6943	6955	6958	6962	6965	6966	6968	6985	7012	7015	7018	7021	7033
7036	7040	7043	7044	7046	7063	7090	7093	7096	7099	7111	7114	7118	7121	7122
7124	7141	7168	7171	7174	7177	7189	7192	7196	7199	7204	7207	7221	7262	7265

```
        7268  7271  7280  7283  7287  7290  7296  7299  7303  7306  7309  7312  7326  7367  7370
        7373  7376  7385  7388  7392  7395  7401  7404  7408  7411  7414  7417  7431  7472  7475
        7478  7481  7490  7493  7497  7500  7506  7509  7513  7516  7519  7522  7536  7577  7580
        7583  7586  7595  7598  7602  7605  7611  7614  7618  7621  7624  7627  7641  7682  7685
        7688  7691  7700  7703  7707  7710  7716  7719  7723  7726  7730  7732  7750  7778  7781
        7785  7788  7801  7804  7808  7811  7812  7814  7832  7860  7863  7867  7870  7883  7886
        7890  7893  7894  7896  7914  7942  7945  7949  7952  7965  7968  7972  7975  7976  7978
        7996  8024  8027  8031  8034  8047  8050  8054  8057  8062  8065  8079  8119  8122  8125
        8128  8137  8140  8144  8147  8153  8156  8160  8163  8165  8168  8182  8222  8225  8228
        8231  8240  8243  8247  8250  8256  8259  8263  8266  8268  8271  8285  8325  8328  8331
        8334  8343  8346  8350  8353  8359  8362  8366  8369  8371  8374  8388  8428  8431  8434
        8437  8446  8449  8453  8456  8462  8465  8469  8472  8474  8477  8491  8531  8534  8537
        8540  8549  8552  8556  8559  8565  8568  8572  8575  8598  8604  8609  8625  8631  8653
        8666  8732  8735  8739  8742  8754  8761  8815  8818  8821  8824  8836  8843  8898  8901
        8904  8907  8919  8926  8980  8983  8987  8990  9002  9009  9063  9066  9070  9073  9085
        9092  9100  9107  9115  9131  9134  9140  9143  9275  9403  9535  9771  9780  9781  9782
        9783  9784  9785  9786  9787  9788  9789
.PAGE    923   969  1255  1347  1399  1445  1491  1537  1583  1629  1676  1815  1860  1905  1950
        1995  2040  2085  2130  2175  2220  2265  2310  2355  2400  2445  2490  2535  2580  2625
        2670  2715  2760  2805  2850  2895  2940  2985  3030  3077  3123  3169  3215  3261  3307
        3353  3399  3445  3490  3535  3580  3625  3670  3715  3760  3805  4742  4781  4820  4859
        4898  4937  5530  5740  5772  5814  5856  5898  5940  5982  6834  6888  6966  7044  7122
        7205  7310  7415  7520  7625  7730  7812  7894  7976  8063  8166  8269  8372  8475
.RADIX  1815  1828  1860  1873  1905  1918  1950  1963  1995  2008  2040  2053  2085  2098  2130
        2143  2175  2188  2220  2233  2265  2278  2310  2323  2355  2368  2400  2413  2445  2458
        2490  2503  2535  2548  2580  2593  2625  2638  2670  2683  2715  2728  2760  2773  2805
        2818  2850  2863  2895  2908  2940  2953  2985  2998  3030  3043  3075  3077  3090  3123
        3136  3169  3182  3215  3228  3261  3274  3307  3320  3353  3366  3399  3412  3445  3458
        3490  3503  3535  3548  3580  3593  3625  3638  3670  3683  3715  3728  3760  3773  3805
        3818  3850
.REM        1
.REPT     750   961   962  2355  3490
.SBTTL    732   744   764   888   900   924   971  1062  1254  1344  1345  1346  1350  1400  1446
        1492  1538  1584  1631  1677  1711  1751  1784  1816  1861  1906  1951  1996  2041  2086
        2131  2176  2221  2266  2311  2356  2401  2446  2491  2536  2581  2626  2671  2716  2761
        2806  2851  2896  2941  2986  3031  3078  3124  3170  3216  3262  3308  3354  3400  3446
        3491  3536  3581  3626  3671  3716  3761  3806  3850  3851  3852  3856  3905  3961  4011
        4065  4102  4145  4189  4227  4270  4314  4349  4385  4420  4457  4488  4520  4564  4631
        4632  4633  4636  4685  4744  4783  4822  4861  4900  4939  4978  5021  5064  5098  5145
        5182  5230  5279  5311  5342  5374  5409  5491  5530  5531  5532  5535  5585  5654  5706
        5740  5774  5816  5858  5900  5942  5984  6026  6074  6075  6076  6084  6212  6280  6337
        6381  6421  6478  6606  6675  6731  6830  6831  6832  6836  6890  6968  7046  7124  7207
        7312  7417  7522  7627  7732  7814  7896  7978  8065  8168  8271  8374  8477  8638  8641
        8678  8679  8680  8684  8765  8847  8930  9013  9145  9146  9147  9150  9228  9277  9325
        9393  9457  9573  9653  9711  9755  9772
.TITLE   720
.WORD    750   751   752   896   916   917   918   919   920   921   932   935   936   937   938
         941   942   943   944   945   946   947   948   949   950   959   961   962   976   977
         978   979   980   981   982   983   987   988   989  1002  1006  1009  1012  1018  1019
        1020  1021  1022  1023  1024  1025  1026  1027  1028  1029  1030  1031  1032  1033  1034
        1035  1036  1037  8658  8661  8673  9143  9144  9226  9303  9308  9602  9649  9685  9746
        9948  9950  9952  9954  9956  9958  9960  9962  9963  9964  9966  9968  9970  9972  9974
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DZKWKA.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZKWK.CMB
RUN-TIME: 129 122 19 SECONDS
RUN-TIME RATIO: 489/271=1.8
CORE USED: 35K (69 PAGES)