LPA/KW11-K

DIAGNOSTIC TEST
MD-11-DRLPG-A

EP-DRLPG-A-DL

FICHE 1 OF 2

MAR 1978
digital
MADE IN USA

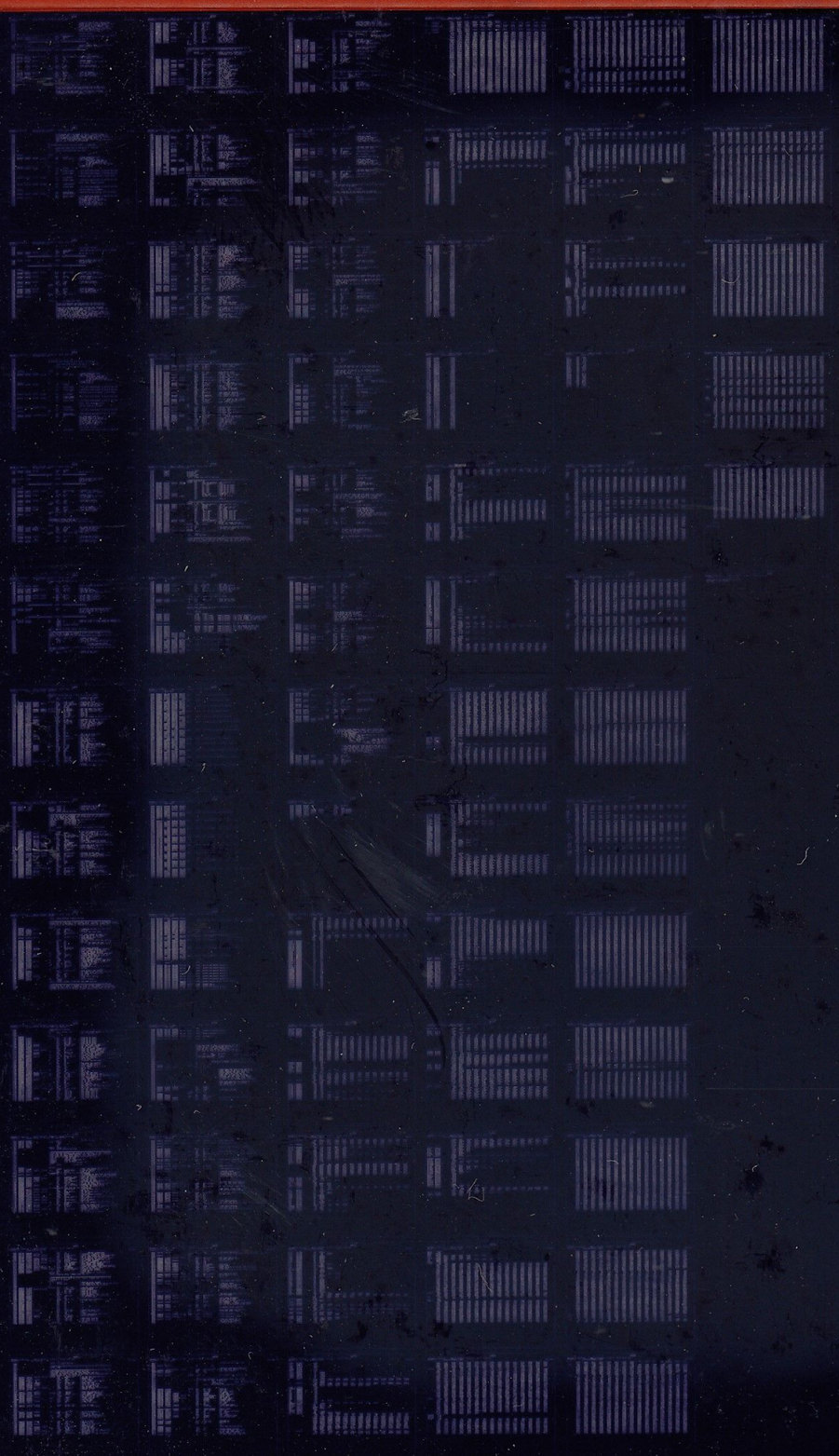
LPA/KW11-K

DIAGNOSTIC TEST
MD-11-DRLPG-A

EP-DRLPG-A-DL

FICHE 2 OF 2

MAR 1978
digital
MADE IN USA



Product Code: MAINDEC-11-DRLPG-A-D

Product Name: LPA/KW11-K DIAGNOSTIC TEST

Date Created: JANUARY 1978

Maintainer: DIAGNOSTIC ENGINEERING

Table of Contents
****

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.

THIS PROGRAM IS A MODIFIED VERSION OF "MD-11-DZKWK-A". IT WAS MODIFIED TO ENABLE THE OPERATOR TO CHECK OUT THE KW11K OPTION WHEN IT IS ON THE LPA11-KX I/O BUS. NO RECABLING IS NEEDED. SOME TEST DONE IN THE ORIGINAL DIAGNOSTIC SUCH AS ARBRITRATION TEST, WERE DELETED AS THEY COULD NOT BE CHECKED. IF THIS DIAGNOSTIC DOESN'T FIND A SUSPECTED PROBLEM, YOU MAY HAVE TO RUN "MD-11-DZKWK-A". YOU SHOULD RUN "MD-11-DRLPA" BEFORE RUNNING THIS DIAGNOSTIC. PLEASE READ SECTION 10.

2.0 REQUIREMENTS
****

2.1 Equipment

1. PDP11 FAMILY COMPUTER with 16K 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.
4. LPA11-KX

2.2 Storage

This program occupies and uses only the lower 16K 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).

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
     176   $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

     EM5                ;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

A HALT MAY OCCUR IF THE PROGRAM DETECTS AN LPA11-KX ERROR.
CHECK THE COMMENTS IN THE LISTING OPPOSITE THE PC HALT.

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

90 SECONDS iterations inhibited - no errors.

375 SECONDS 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-11/34.

8.4 USER LINK TO I/O DEVICE

A SPECIAL USER LINK HAS BEEN PROVIDED IN ORDER FOR THE

KO1

OPERATOR TO EXAMINE OR MODIFY LOCATIONS ON THE LPA11-KX I/O BUS. (NOTE: THIS CANNOT BE DONE DIRECTLY.)

SEQ 0010

PROCEDURE:

1) START THE PROCESSOR AT LOCATION $UTK:

2) THE DIALOG TO EXAMINE A LOCATION IS AS FOLLOWS:

```
E OR D              "E"
DEVICE ADDRS=       "OCTAL ADDRS"
XXXXXX
```

WHERE XXXXXX IS THE CONTENTS OF THE SPECIFIED LOC.

3) THE DIALOG TO MODIFY A LOCATION IS AS FOLLOWS:

```
E OR D              "D"
DATA=               "DATA TO BE DEPOSITED"
```

4) THE PROGRAM WILL STAY IN THIS LOOP UNTIL THE OPERATOR IS FINISHED. AT THIS TIME THE PROCESSOR SHOULD BE HALTED.

NOTE: THE OPERATORS RESPONSE IS ENCLOSED IN QUOTES.

9.0 PROGRAM DESCRIPTION
****

9.1 Logic Tests

A complete description of each test is included withing the listing before each test.

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.

Logic tests (L + S at 200) should be run first.

9.2.3 LS220 "SCHMITT TRIG 3" in, "ST3 OUT" Tests SEQ 0012

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 tne 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.

10.0 LPA11 (SYSTEM) DIAGNOSTIC SUMMARY

DIAGNOSTICS FOR THE LPA11 ARE WRITTEN AT THREE LEVELS: (1) TOTAL PDP-11 SYSTEM, (2) LPA11 SYSTEM; AND, (3) LPA11 OPTIONS.

LEVEL 1, IS DESIGNED TO ISOLATE A FAILURE TO THE LPA11 SYSTEM. ALL OPTIONS ON THE PDP-11 ARE EXERCISED.

LEVEL 2 DIAGNOSTICS ISOLATE A FAILURE TO THE INDIVIDUAL OPTION WITHIN THE LPA11. THE LEVEL 2 DIAGNOSTIC IS MD-11-DRLPA. WHEN THE USER RUNS DRLPA HE CAN GENERALLY TELL WHICH OPTION DIAGNOSTIC (LEVEL 3) TO RUN NEXT. M8254 AND M8200-YC ERRORS MAY "LOOK" ALIKE AND DRLPA MAY NOT BE ABLE TO DISTINGUISH BETWEEN THEM. ARBITRATION ERRORS WILL NOT BE DETECTED BY THIS DIAGNOSTIC.

LEVEL THREE DIAGNOSTICS AID IN DETERMINING IF THE ERROR WAS IN FACT ON THE OPTION THE DRLPA SPECIFIED. THE USER MAY "LOOP" ON THE ERROR. WITHIN LEVEL THREE, THERE ARE TWO GROUPS OF DIAGNOSTICS. THE FIRST GROUP REQUIRES NO "EXTRA" WORK BY THE USER IN ORDER TO RUN. GROUP "A" DIAGNOSTICS DO NOT CHECK ARBITRATION, AND REQUIRE EXTRA TIME FOR EXECUTION. THE SECOND GROUP (GROUP "B") REQUIRES THAT THE USER RECONFIGURE THE PDP-11 SYSTEM. THIS RECONFIGURATION INVOLVES CABLING THE UNIBUS TO THE LPA'S I/O BUS.

THE DIAGNOSTIC FOR THE M8254 FALLS INTO THE GROUP "B" CATAGORY.

THE LPA11-KX DIAGNOSTIC KIT WILL INCLUDE:

SEQ 0014

OPTION	GROUP	DIAG. #	DIAG. TITLE
LPA11-KX	LEVEL 2	MD-11-DRLPA	LPA11-K SYSTEM DIAG.
M8254	"B"	MD-11-DRM8A	M8254 (JPBM) DIAG.
AA11-K	A	MD-11-DRLPB	AA11-K DIAG.
	B	MD-11-DZAAC	AA11-K DIAG.
AR11	A	MD-11-DRLPC	LPA/AR11 DIAG. #1
	A	MD-11-DRLPD	LPA/AR11 DIAG. #2
	A	MD-11-DRLPE	LPA/AR11 DIAG. #3
	B	MD-11-DZARA	AR11 DIAG. #1
	B	MD-11-DZARB	AR11 DIAG. #2
	B	MD-11-DZARC	AR11 DIAG. #3
DR11-K	A	MD-11-DRLPF	LPA/DR11-K DIAG.
	B	MD-11-DZDRG	DR11-K DIAG.
KW11-K	A	MD-11-DRLPG	LPA/KW11-K DIAG.
	B	MD-11-DZKWK	KW11-K DIAG.
LPS11	A	MD-11-DRLPH	LPA/LPS11 DIAG. #1
	A	MD-11-DRLPI	LPA/LPS11 DIAG. #2
	A	MD-11-DRLPJ	LPA/LPS11 DIAG. #3
	B	MD-11-DZLPC	LPS11 DIAG. #1
	B	MD-11-DZLPD	LPS11 DIAG. #2
	B	MD-11-DZLPI	LPS11 DIAG. #3
AD11-K	A	MD-11-DRLPK	LPA/AD11-K DIAG.
	B	MD-11-DZADL	AD11-K DIAG.
M8200-YC	B	MD-11-DZLPL	LPA/M8200-YC BASIC MICRO-CPU R/W TEST
	B	MD-11-DZLPM	LPA/M8200-YC JMP+ROM READ TEST

```
                        .REM    [

                                        LPA.MAC

WELCOME, THIS DIAGNOSTIC IS ONE IN A SERIES OF DIAGNOSTIC
DESIGNED IN ORDER TO AID YOU IN TESTING THE LPA-11XX OPTION.
I HOPE THAT YOU HAVE READ THE DOCuMENTATION SECTION OF THIS
DIAGNOSTIC.  IF YOU HAVE,YOU KNOW ABOUT ALL OF THE DIAGNOSTICS
THAT ARE AVAILIBLE FOR TESTING THE LPA SYSTEM.

                                        GOOD LUCK !

[
.GLOBL  DRLPX2
```

```
                                .REM    !

                                THIS IS A LIST OF TESTS DELETED FROM THIS DIAGNOSTIC.
                                THESE TEST COULD NOT BE DONE THROUGH THE LPA-11.

                                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'STATUS REGISTER
                                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 THAT A'S COUNT REGISTER IS CLEARED BY INIT
                                TEST THAT CLOCK A WILL INTR. AND TO THE 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 STI WILL CAUSE CLOCK A TO INTER.
                                TEST THAT CLOCK A OVERFLOW WILL CAUSE AN INTR.
                                TEST THAT A CLOCK A COUNTER BUFFER WILL CAUSES AN INTR.
                                TEST THAT CLOCK B WILL INTR. AND TO THE RIGHT VECTOR
                                TEST THAT CLOCK B 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 CLOCKB OVERFLOW WILL CAUSE AN INTR.
                                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 REPEATIBILITY AT 1MHZ RATE
                                TEST CLOCK B'S REPEATIBILITY AT 100KHZ 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
                                TEST THAT "INIT" CLEARS B'S 100KHZ DIVIDE BY 10 CHIPS
                                    !


                                .TITLE  MAINDEC-11-DRLPG-A
                                ;*COPYRIGHT (C) 1978
                                ;*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-C2), SEPT 14, 1976.
                                ;*
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

                                        .=0
                    000000      ;*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
000174  000000     DISPREG:   .WORD  0             ;;SOFTWARE DISPLAY REGISTER
000176  000000     SWREG:     .WORD  0             ;;SOFTWARE SWITCH REGISTER
                    000200              .=200
000200  000137 001730                  JMP     @#START         ;GO TO STARTING ADDRESS OF PROGRAM

000204  000137 002434                  JMP     @#RSTART        ;GO TO RESTART ADDRESS.
000210  000137 023376                  JMP     @#LS210         ;GO TO SPECIAL TEST #1.
000214  000137 023564                  JMP     @#LS214         ;GO TO SPECIAL TEST #2.
000220  000137 024000                  JMP     @#LS220         ;GO TO SPECIAL TEST #3.
000224  000137 024160                  JMP     @#LS224         ;GO TO SPECIAL TEST #4.
000230  000137 024372                  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
                    000006      SP=     %6              ;;STACK POINTER
```

```
                    000007          PC=     %7              ;;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
000234                          $SVPC=.                 ;SAVE PC
000046                          .=46
000046  023344                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                          .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
000234                          .=$SVPC                 ;; RESTORE PC
001000                          .=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
                        200     ;;FOR APT START UP
                        .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                        $APTHDR ;;POINT TO APT HEADER BLOCK
                        .=.$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  001202                  $MBADR: .WORD   $MAIL       ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000002                  $TSTM:  .WORD   2           ;;RUN TIM OF LONGEST TESt
001006  000120                  $PASTM: .WORD   120         ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000120                  $UNITM: .WORD   120         ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .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      000         $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135      000         $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154      000         $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155      002         $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156      012         $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157      000         $TPFLG: .BYTE   0               ;;"TERminAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REGO) WAS OBTAINED
001162  000000          $REGO:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $TMPO:  .WORD   0               ;;USER DEFINED
001166  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001170  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001172  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001176      077         $QUES:  .ASCII  /?/             ;;QUESTION MARK
001177      015         $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001200  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001202                  $MAIL:                          ;;APT MAILBOX
001202  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
```

```
                                ;
                                $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
                                $TESTN: .WORD   ATESTN  ;;TEST NUMBER
                                $PASS:  .WORD   APASS   ;;PASS COUNT
                                $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
                                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
                                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
                                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
                                $ETABLE:                ;;APT ENVIRONMENT TABLE
                                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
                                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
                                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
                                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
                                $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
                                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                              900 NSEC CORE=001
                                ;*                              300 NSEC BIPOLAR=002
                                ;*                              500 NSEC MOS=003
                                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                .*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
                                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
                                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
                                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
                                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
                                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
                                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
                                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
                                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
                                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
                                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
                                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
                                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
                                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
                                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
                                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
                                $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
                                $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
                                $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
                                $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
```

```
                                $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
                                $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
                                $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
                                $DDW15: .WORD   ADDW15  ;;DEVICE DEScRIPTOR WORD#15


001326                          $ETEND:


001326  170404                  ASR:    170404          ;/CLOCK A STATUS REGISTER.
001330  170406                  ABR:    170406          ;/CLOCK A BUFFER REGISTER.
001332  170430                  ACR:    170430          ;/CLOCK A COUNT REGISTER.

001334  170432                  BSR:    170432          ;/CLOCK B STATUS REGISTER.
001336  170434                  BBR:    170434          ;/CLOCK B BUFFER REGISTER.
001340  170436                  BCR:    170436          ;/CLOCK B COUNT REGISTER.

001342  000344                  AVECT:  344             ;/CLOCK A INTR. VECTOR ADDR.
001344  000346                  AVECP2: 346             ;/CLOCK A INTR. STATUS WORD.

001346  000364                  BVECT:  364             ;/CLOCK B INTR. VECTOR ADDR.
001350  000366                  BVECT2: 366             ;/CLOCK B INTR. STATUS WORD.

001352  000006                  APRITY: 6               ;/PRIORITY LEVEL OF CLOCK A.
001354  000006                  BPRITY: 6               ;/PRIORITY LEVEL OF CLOCK B.
001356  000000                  $TMDAT: 0
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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

001360                          $ERRTB:

                                ;ITEM   1

001360  030112                          EM1                     ;CLOCK A SR FUNCTION ERROR
001362  030777                          DH1                     ;ERRPC    ASR     WAS     S/B
001364  031612                          DT1                     ;$ERRPC,ASR,$BDDAT,$GDDAT
001366  032014                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   2

001370  030145                          EM2                     ;CLOCKA SR DATA ERROR
001372  030777                          DH1                     ;ERRPC    ASR     WAS     S/B
001374  031612                          DT1                     ;$ERRPC,ASR,$BDDAT,$GDDAT
001376  032014                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   3

001400  030174                          EM3                     ;CLOCKA BR DATA ERROR
001402  031035                          DH3                     ;ERRPC    ABR     WAS     S/B
001404  031624                          DT3                     ;$ERRPC,ABR,$BDDAT,$GDDAT
001406  032014                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   4

001410  030223                          EM4                     ;CLOCKA CR DATA ERROR
001412  031073                          DH4                     ;ERRPC    ACR     WAS     S/B
001414  031636                          DT4                     ;$ERRPC,ACR,$BDDAT,$GDDAT
001416  032014                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   5

001420  030252                          EM5                     ;CLOCK B SR DATA ERROR
001422  031131                          DH5                     ;ERRPC    BSR     WAS     S/B
001424  031650                          DT5                     ;$ERRPC,BSR,$BDDAT,$GDDAT
001426  032014                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
```

```
                                        ;ITEM   6
001430  030301                                  EM6             ;CLOCK B BR DATA ERROR
001432  031167                                  DH6             ;ERRPC    BBR       WAS       S/B
001434  031662                                  DT6             ;$ERRPC,BBR,$BDDAT,$GDDAT
001436  032014                                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM

                                        ;ITEM   7
001440  030330                                  EM7             ;CLOCK B CR DATA ERROR
001442  031225                                  DH7             ;ERRPC    BCR       WAS       S/B
001444  031674                                  DT7             ;$ERRPC,BCR,$BDDAT,$GDDAT
001446  032014                                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM

                                        ;ITEM   10
001450  030357                                  EM10            ;DUAL ADDRESS ERROR
001452  031263                                  DH10            ;ERROR     GOOD     BAD      GOOD     DATA READ FROM
                                                                ;   PC      ADDR     ADDR     DATA     DUAL ADDRESS
001454  031706                                  DT10            ;$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT
001456  032014                                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM

                                        ;ITEM   11
001460  030404                                  EM11            ;CLOCK A COUNT ERROR
001462  031073                                  DH4             ;ERRPC    ACR    WAS    S/B
001464  031636                                  DT4             ;$ERRPC, ACR, $BDDAT, $GDDAT
001466  032014                                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM

                                        ;ITEM   12
001470  030433                                  EM12            ;CLOCK A COUNT FUNCTION ERROR
001472  031422                                  DH12            ;ERRPC    ASR
001474  031722                                  DT12            ;ERRPC, ASR
001476  032014                                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM

                                        ;ITEM   13
001500  032014                                  DF0             ;ERROR 13 DOES NOT EXSIST.
001502  032014                                  DF0             ;IT WOULD BE BAD LUCK.
001504  032014                                  DF0
001506  032014                                  DF0

                                        ;ITEM   14
001510  030473                                  EM14            ;CLOCK B COUNT FUNCTION ERROR
001512  031441                                  DH14            ;ERRPC    BSR
```

```
001514  031730                  DT14            ;$ERRPC, BSR
001516  032014                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   15

001520  030533                  EM15            ;CLOCK B COUNT ERROR
001522  031225                  DH7             ;ERRPC   CSR   WAS   S/B
001524  031674                  DT7             ;$ERRPC, BCR, $BDDAT, $GDDAT
001526  032014                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   16

001530  030562                  EM16            ;CLOCK A INTERRUPT ERROR
001532  031422                  DH12            ;ERRPC   ASR
001534  031722                  DT12            ;$ERRPC, ASR
001536  032014                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   17

001540  030615                  EM17            ;CLOCK B INTERRUPT ERROR
001542  031441                  DH14            ;ERRPC   BSR
001544  031730                  DT14            ;$ERRPC, BSR
001546  032014                  DF0


                        ;ITEM   20

001550  030650                  EM20            ;CLOCK A REPEATABILITY ERROR
001552  031457                  DH20            ;ERROR   ASR   2ND CNT 1ST CNT
001554  031612                  DT1             ;$ERRPC, ASR, $BDDAT, $GDDAT
001556  032014                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   21

001560  030404                  EM11            ;CLOCK A COUNT ERROR
001562  031073                  DH4             ;ERROR   ASR   2ND CNT 1ST CNT
001564  031736                  DT21            ;$ERRPC, ASR, $BDDAT, $GDDAT
001566  032014                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   22

001570  030404                  EM11            ;CLOCK A COUNT ERROR
001572  031073                  DH4             ;ERRPC   ASR   WAS   S/B
001574  031750                  DT22            ;$ERRPC, ACR, $BDDAT, $TMPD
001576  032014                  DF0             ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   23

001600  030707                  EM23            ;CLOCK B REPEATABILITY ERROR
```

```
001602  031521                  DH23                    ;ERROR   ASR   2NDCNT  1STCNT
001604  031612                  DT1                     ;$ERRPC, ASR, $BDDAT, $GDDAT
001606  032014                  DF0                     ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   24

001610  030533                  EM15                    ;CLOCK B COUNT ERROR
001612  031225                  DH7                     ;ERRPC   BCR   WAS   S/B
001614  031762                  DT24                    ;$ERRPC, BCR, $GDDAT, $TMP0
001616  032014                  DF0                     ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   25

001620  030533                  EM15                    ;CLOCK B COUNT ERROR
001622  031225                  DH7                     ;ERRPC   BCR     WAS     S/B
001624  031774                  DT25                    ;$ERRPC,BCR,$BDDAT,$TMP0
001626  032014                  DF0                     ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ITEM   26

001630  030746                  EM26    ;CLOCK ADDRESSING ERRROR
001632  031563                  DH26    ;ERRPC   CLOCK ADDR.
001634  032006                  DT26    ;$ERRPC,$TMP0
001636  032014                  DF0                     ;ALL NUMBERS ARE IN OCTAL FORM


                        ;ADDRESS OF KMC-11 OF LPA-11    THE ADDR FOR KMAD0 MAY BE
                        ;                               CHANGED BY THE USER TO rEFLECT
                        ;                               A DIFFERENT KMC-11 ADDR. THE
                        ;                               REST OF THE ADDRESSES WILL
                        ;                               BE CHANGED BY THE PROGRAM.
                        ;

001640                  LPCI:
001640  170460          KMAD0:  .WORD   170460          ;BASE KMC ADDR. MAY BE PATCHED BY USER.

001642                  LPMR:
001642  170461          KMAD1:  .WORD   170460+1                ;>DO NOT        <;KMC-CSR ADDR
001644                  LPC0:
001644  170462          KMAD2:  .WORD   170460+2                ;>PATCH         <;
001646                  LPS0:
001646  170463          KMAD3:  .WORD   170460+3                ;>THIS AREA     <
001650                  LPADL:
001650  170464          KMAD4:  .WORD   170460+4                ;
001652                  LPADH:
001652  170465          KMAD5:  .WORD   170460+5                ;>DO NOT        <
001654                  LPMS1:
001654  170466          KMAD6:  .WORD   170460+6                ;>PATCH         <
001656                  LPMS2:
```

```
  618  001656  170467                KMAD7:  .WORD   170460+7                ;>THIS AREA     <
  619
  620  001660  000344                VECTOR: .WORD   AVECT1&777              ;BASE VECTOR OF KMC
  621  001662  000350                VECTPS: .WORD   4+AVECT1&777            ;VECOTR ADDR.+2
  622
  623  001664  000004                VERSN:  .WORD   4                       ;CURRENT VERSION NUMBER OF MICROCODE.
  624
  625  001666  000000                .DVLS:  .WORD   0                       ;/DEVICE LIST OF I/O ADDR. DEFINED
  626  001670  000020                        .BLKW   16.                     ;/BY INIT.
  627
  628                                        .SBTTL  PROGRAM START
```

```
                                                ;
001730                                          START:
                                                .SBTTL  INITIALIZE THE COMMON TAGS
                                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001730  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001734  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001736  022706  001140                          CMP     #SWR,R6 ;;DONE?
001742  001374                                  BNE     .-6             ;;LOOP BACK IF NO
001744  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                                ;;INITIALIZE A FEW VECTORS
001750  012737  025762  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001756  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001764  012737  025214  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001772  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002000  012737  030026  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002006  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
002014  012737  027650  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002022  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002030  005037  001166                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002034  005037  001170                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002040  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002046  012737  002046  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002054  012737  002054  001110                  MOV     #.,$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.
002062  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002066  012737  002122  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002074  012737  177570  001140                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002102  012737  177570  001142                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002110  022777  177777  177022                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002116  001012                                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
002120  000403                                  BR      65$             ;;BRANCH IF NO TIMEOUT
002122  012716  002130          64$:            MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002126  000002                                  RTI
002130  012737  000176  001140  65$:            MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002136  012737  000174  001142                  MOV     #DISPREG,DISPLAY
002144  012637  000004          66$:            MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
                                                ;
002150  005037  001210                          CLR     $PASS           ;;CLEAR PASS COUNT
002154  132737  000200  001223                  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002162  001403                                  BEQ     67$             ;;YES,USE NON-APT SWITCH
002164  012737  001224  001140                  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002172                          67$:
002172  005737  000042                          TST     @#42            ;IF RUNNING UNDER ACT-
002176  001015                                  BNE     10$             ;NO TYPEOUT.
                                                ;
002200  104401  002206                          TYPE    ,69$            ;;TYPE ASCIZ STRING
002204  000412                                  BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <15><12><12>#MD-11-DRLPG-A#<15><12>
002232                          68$:
                                                ;
002232  013737  001256  001326  10$:             MOV     $BASE,ASR
002240  012737  000001  001212                  MOV     #1,$DEVCT
```

```
                                        CLR     $PASS
                                        ;
                                        ;THIS SECTION OF CODE HANDLES INITIALIZING LPA-11 FUNCTIONS
                                        ;
                                        MOV     R0,-(SP)
                                        MOV     R1,-(SP)
                                        MOV     KMADD,R0        ;GET KMC-11 ADDRESS.
                                        MOV     #KMAD1,R1       ;GET ADDR. OF ADDR. LIST.

                                70$:    INC     R0              ;UPDATE ADDR.
                                        MOV     R0,(1)+         ;WRITE ADDR.
                                        CMP     R1,#KMAD7+2     ;DONE ALL ADDRESSES?
                                        BNE     70$             ;NO - DO NEXT ADDR.
                                        CLR     .DVLs           ;CLR ADDR. LIST.
                                        MOV     (SP)+,R1
                                        MOV     (SP)+,R0

                                LOOP:
                                        CLR     R0
                                1$:     INC     R0              ;DELAY SOME TIME SO THAT FIRST RESET
                                        BNE     1$              ;INSTR. WON'T CLOBBER TYPEOUT.
                                        MOV     ASR,R0          ;NOW WE'RE GONNA FIX
                                        ADD     #2,R0           ;ALL CLOCK ADDRESSES BASED ON ASR.
                                        MOV     R0,ABR
                                        ADD     #22,R0
                                        MOV     R0,ACR
                                        ADD     #2,R0
                                        MOV     R0,BSR
                                        ADD     #2,R0
                                        MOV     R0,BBR
                                        ADD     #2,R0
                                        MOV     R0,BCR

                                        MOV     AVECT,R0        ;NOW FIX VECTOR ADDRESSES
                                        ADD     #2,R0           ;BASED ON AVECT.
                                        MOV     R0,AVECP2
                                        ADD     #16,R0
                                        MOV     R0,BVECT
                                        ADD     #2,R0
                                        MOV     R0,BVECT2

                        001354          MOV     APRITY,BPRITY   ;FIX CLK B'S PRIORITY BASED ON A'S.
                                RSTART: MOV     #STACK,SP
                                        MOV     #340,-(SP)      ;SET PROCESSOR PRIORITY TO 7.
                                        MOV     #1$,-(SP)
                                        RTI
                                1$:

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

```
                                        ;;*****************************************************************
                                        ;*TEST 1         *TEST THE ADDRESSABILITY OF CLOCK ADDRESS
                                        ;*
                                        ;*"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
                                        ;*
                                        ;*****************************************************************
002452  000240                  TST1:   NOP
002454  012737  000050  001166          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
002462  012737  002470  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
002470  112737  000001  001102  1$:     MOVB    #1,$TSTNM
002476  112737  000001  001206          MOVB    #1,$TESTN


                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.

                                        ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.

                                        ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.


                                        ;*****************************************************************
                                        ;*TEST 2         *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)
                                        ;*
                                        ;*

                                        ;;*****************************************************************
002564  000004                    TST2:   SCOPE
002566  012737  000050  001166            MOV     #50,$TIMES      ;;DO 50 ITERATIONS

                                                                    ;WRITE INTO PRESET BUFFER.
                                                                    ;SET $GDDAT FOR ERROR TYPEOUT.
                                                                    ;READ THE PRESET BUFFER.
002574  012737  001416  001124            MOV     #1416,$GDDAT

                                ;*          MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*          MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
002622  005737  001126                    TST     $BDDAT
002626  001001                            BNE     1$              ;IF ANY DATA WAS READ BACK, WE WILL
                                                                    ;BR PAST THE ERROR CALL.


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

002630  104003                            ERROR   3               ;UNABLE TO LOAD AND READ BACK
                                                                    ;FROM THE BUFFER REGISTER CLOCK A.

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


002632                          1$:
```

```
                        ;;**********************************************************************
                        ;*TEST 3         *TEST THAT CLOCK A BUFFER CAN BE WRITTEN TO A ZERO
                        ;*
                        ;*THE LAST TEST WROTE 1'S INTO CLOCK A'S BUFFER. IN THIS
                        ;*TEST WE TRY TO WRITE ALL ZEROS.
                        ;*
                        ;*SIGNALS - SAME AS LAST TEST. SUSPECT F/F OR DATA GATE STUCK OPEN
                        ;*
                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                        ;*
                        ;*
                        ;;**********************************************************************
002632  000004         TST3:   SCOPE
002634  012737  000050  001166          MOV     #50,$TIMES      ;;DO 50 ITERATIONS

002642  005037  001124          CLR     $GDDAT          ;INDICATE WE EXPECT 0'S.
                                                        ;CLEAR BUFFER REGISTER.
                                                        ;READ CLOCK A'S BUFFER REGISTER
                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
002666  005737  001126          TST     $BDDAT
002672  001401                  BEQ     1$              ;SHOULD BE CLEAR - IF CLEAR - NEXT TEST.

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

002674  104003                  ERROR   3               ;CLEAR INTR. FAILED TO CLEAR CLOCK A'S
                                                        ;BUFFER REGISTER.

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

002676                  1$:

                        ;;**********************************************************************
                        ;*TEST 4         *TEST THAT CLOCK A'S STATUS CAN BE WRITTEN AND READ
                        ;*
                        ;*NOW THAT WE CAN WRITE INTO THE BUFFER REGISTER, WE'RE GOING TO TRY
                        ;*WRITING INTO THE STATUS REGISTER AND READ IT BACK.
                        ;*
                        ;*NEW SIGNALS: ["BA03" L + "BA02" L + "BA01" L]="LD STAT A" L
                        ;*      "DATA OUT LO" L + "DATA OUT HI" L + "LD STATA" L = "LD STAT A HI" H
                        ;*      + "LD STATA LO H"
                        ;*FOR READ BACK: "RD STATA" L
                        ;*
                        ;*NO ATTEMPT MADE TO VERIFY THAT CORRECT DATA CAME BACK, BUT
                        ;*JUST THAT SOME DATA CAME BACK.
                        ;*
```

```
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEV ENABLE (1)" 2 OCCURANCES PER PASS
                                ;*
                                ;*

                                ;;*****************************************************************
002676  000004                 TST4:   SCOPE
002700  012737  000050  001166         MOV     #50,$TIMES      ;;DO 50 ITERATIONS

002706  012737  001416  001124         MOV     #1416,$GDDAT    ;LOAD $GDDAT WITH S/B.
                                                               ;LOAD STATUS REGISTER.
                                                               ;READ BACK STATUS REGISTER

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
002734  005737  001126                 TST     $BDDAT
002740  001001                         BNE     1$              ;IF ANY BITS RETURNED - NO ERROR.

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

002742  104002                         ERROR   2               ;ERROR-UNABLE TO LOAD AND READ BACK
                                                               ;STATUS REGISTER OF CLOCK A.

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

002744                          1$:                            ;CLEAR CLOCKA'S STATUS REG.
002744  005037  001124                 CLR     $GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;;*****************************************************************
                                ;*TEST 5         *TEST THAT CLOCK B'S STATUS REGISTER CAN BE WROTE/READ
                                ;*
                                ;*NEW SIGNALS: ["BA03" H + "BA02" L + "BA01" L]="LD STATB" L*"RD STAT B" L
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEV ENABLE (1)" 2 OCCURANCES PER PASS
                                ;*
                                ;*

                                ;;*****************************************************************
002760  000004                 TST5:   SCOPE
002762  012737  000050  001166         MOV     #50,$TIMES      ;;DO 50 ITERATIONS

002770  012737  001016  001124         MOV     #1016,$GDDAT    ;USE 1016 AS PATTERN, PUT IN $GDDAT.
                                                               ;LOAD B'S STAT REG.
                                                               ;READ BACK THE STATUS REG.

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
```

```
003016  005737  001126          TST     $BDDAT
003022  001001                  BNE     1$              ;IF ANY BITS CAME BACK, SUBTEST OK.


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


003024  104005                  ERROR   5               ;ERROR-COULD NOT WRITE/READ BACK
                                                        ;CLOCK B'S STATUS REGISTER.


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


003026                  1$:


                        ;;**********************************************************************
                        ;*TEST 6         *TEST THAT CLOCK B'S BUFFER REGISTER CAN BE WROTE/READ
                        ;*
                        ;*NEW SIGNALS: ["BA03" H + "BA02" L + "BA01" H]="LD BUFF B" L*"RD BUFF B" L
                        ;*
                        ;* PROBABLE SYNC POINT FOR THIS TESt:: "DEV ENABLE (1)" 2 OCCURANCES PER PASS
                        ;*
                        ;**********************************************************************
003026  000004          TST6:   SCOPE
003030  012737  000050  001166          MOV     #50,$TImES      ;;DO 50 ITERATIONS

003036  012737  000370  001124          MOV     #370,$GDDAT     ;USE PATTERN "370", PUT IN $GDDAT.
                                                                ;WRITE INTO CLOCK B'S BUFFER REGISTER.
                                                                ;READ IT BACK.

                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                        ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
003064  005737  001126          TST     $BDDAT
003070  001001                  BNE     1$              ;IF ANY BITS COME BACK-SUBTEST OK.


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


003072  104006                  ERROR   6               ;ERROR-FAILED TO WRITE/READ CLOCKB'S
                                                        ;BUFFER REGISTER.


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


003074                  1$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 7         *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
003074  000004                    TST7:   SCOPE
003076  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE STATUS REGISTER.
                                                                    ;/SET BIT 15.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE STATUS REGISTER.
003104  012737  100000  001124            MOV     #BIT15,$GDDAT

                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003132  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;/DID BIT 15 AND ONLY BIT 15 SET?
003140  001402                             BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003142  104002                            ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                    ;/BIT 15 FAILED TO BIT SET.

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

003144  000416                            BR      2$              ;/BR TO END SUBTEST.
003146                          1$:                               ;/TRY CLEARING BIT 15.
003146  005037  001124                     CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                    ;/NOW READ IT BACK.

                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003172  005737  001126                     TST     $BDDAT
003176  001401                             BEQ     2$              ;/IF ZERO-NO ERROR!

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

003200  104002                            ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                    ;/BIT 15 FAILED TO cLEAR.

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

```
  1038  003202                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************
                                        ;*TEST 10        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************
003202  000004                    TST10:  SCOPE
003204  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE STATUS REGISTER.
                                                                    ;/SET BIT 14.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE STATUS REGISTER.
003212  012737  040000  001124            MOV     #BIT14,$GDDAT
                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003240  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 14 AND ONLY BIT 14 SET?
003246  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003250  104002                           ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 14 FAILED TO BIT SET.

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

003252  000416                           BR      2$              ;/BR TO END SUBTEST.
003254                          1$:                             ;/TRY CLEARING BIT 14.
003254  005037  001124                    CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003300  005737  001126                    TST     $BDDAT
003304  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

003306  104002                           ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 14 FAILED TO CLEAR.

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

```
  1093  003310                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 11       *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003310  000004                    TST11:  SCOPE
003312  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                  ;/CLEAR THE STATUS REGISTER.
                                                                  ;/SET BIT 13.
                                                                  ;/SET FOR ERROR TYPEOUT S/B.
                                                                   ;/READ THE STATUS REGISTER.
003320  012737  020000  001124            MOV     #BIT13,$GDDAT

                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003346  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 13 AND ONLY BIT 13 SET?
003354  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003356  104002                           ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                  ;/BIT 13 FAILED TO BIT SET.

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

003360  000416                           BR      2$              ;/BR TO END SUBTEST.
003362                          1$:             ;/TRY CLEARING BIT 13.
003362  005037  001124                   CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                  ;/NOW READ IT BACK.

                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003406  005737  001126                   TST     $BDDAT
003412  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

003414  104002                           ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                  ;/BIT 13 FAILED TO CLEAR.

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

F04


```
  1148  003416                    2$:
```

```
1149                                    ;/*
1150                                    ;;**********************************************************************
1151                                    ;*TEST 12        *TEST THAT CLOCK A STATUS REGISTER BIT 9 CAN BE SET AND CLEARED
1152                                    ;*
1153                                    ;*CLOCK A STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
1154                                    ;*F/FS OR GATES
1155                                    ;*
1156                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
1157                                    ;*
1158                                    ;*
1159                                    ;;**********************************************************************
1160  003416  000004                    TST12:  SCOPE
1161  003420  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1162                                                                    ;/CLEAR THE STATUS REGISTER.
1163                                                                    ;/SET BIT 9.
1164                                                                    ;/SET FOR ERROR TYPEOUT S/B.
1165                                                                     ;/READ THE STATUS REGISTER.
1166  003426  012737  001000  001124            MOV     #BIT9,$GDDAT
1167
1168                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
1169
1170                                    ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
1171  003454  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 9 AND ONLY BIT 9 SET?
1172  003462  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
1173
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

1177  003464  104002                           ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
1178                                                                    ;/BIT 9 FAILED TO BIT SET.
1179
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

1183  003466  000416                           BR      2$              ;/BR TO END SUBTEST.
1184  003470                          1$:                             ;/TRY CLEARING BIT 9.
1185  003470  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
1186                                                                    ;/NOW READ IT BACK.
1187
1188                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
1189
1190                                    ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
1191  003514  005737  001126                   TST     $BDDAT
1192  003520  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!
1193
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

1197  003522  104002                           ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
1198                                                                    ;/BIT 9 FAILED TO CLEAR.
1199
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

H04


```
  1203  003524                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 13       *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003524  000004                          TST13:  SCOPE
003526  012737  000100  001166                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 8.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
003534  012737  000400  001124                  MOV     #BIT8,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003562  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 8 AND ONLY BIT 8 SET?
003570  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003572  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 8 FAILED TO BIT SET.

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

003574  000416                          BR      2$              ;/BR TO END SUBTEST.
003576                          1$:                             ;/TRY CLEARING BIT 8.
003576  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003622  005737  001126                  TST     $BDDAT
003626  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

003630  104002                          ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 8 FAILED TO CLEAR.

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

```
  1258  003632                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 14        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003632  000004                        TST14:  SCOPE
003634  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 7.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
003642  012737  000200  001124                MOV     #BIT7,$GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003670  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
003676  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003700  104002                               ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 7 FAILED TO BIT SET.

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

003702  000416                               BR      2$              ;/BR TO END SUBTEST.
003704                          1$:                                   ;/TRY CLEARING BIT 7.
003704  005037  001124                       CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003730  005737  001126                       TST     $BDDAT
003734  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

003736  104002                               ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 7 FAILED TO CLEAR.

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

```
 1313  003740                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 15        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
003740  000004                        TST15:  SCOPE
003742  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 6.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
003750  012737  000100  001124                MOV     #BIT6,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
003776  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
004004  001402                              BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004006  104002                              ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 6 FAILED TO BIT SET.

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

004010  000416                              BR      2$              ;/BR TO END SUBTEST.
004012                          1$:                                 ;/TRY CLEARING BIT 6.
004012  005037  001124                      CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004036  005737  001126                      TST     $BDDAT
004042  001401                              BEQ     2$              ;/IF ZERO-NO ERROR!

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

004044  104002                              ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 6 FAILED TO CLEAR.

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

```
 1368  004046                        2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 16        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
004046  000004                    TST16:  SCOPE
004050  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE STATUS REGISTER.
                                                                    ;/SET BIT 5.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE STATUS REGISTER.
004056  012737  000040  001124            MOV     #BIT5,$GDDAT
                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004104  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
004112  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004114  104002                           ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 5 FAILED TO BIT SET.

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

004116  000416                           BR      2$              ;/BR TO END SUBTEST.
004120                          1$:             ;/TRY CLEARING BIT 5.
004120  005037  001124                   CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*          MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*          MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004144  005737  001126                   TST     $BDDAT
004150  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

004152  104002                           ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 5 FAILED TO CLEAR.

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

```
  1423  004154                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 17       *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
004154  000004                          TST17:  SCOPE
004156  012737  000100  001166                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 3.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
004164  012737  000010  001124                  MOV     #BIT3,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004212  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
004220  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004222  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 3 FAILED TO BIT SET.

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

004224  000416                          BR      2$              ;/BR TO END SUBTEST.
004226                          1$:                             ;/TRY CLEARING BIT 3.
004226  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004252  005737  001126                  TST     $BDDAT
004256  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

004260  104002                          ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 3 FAILED TO CLEAR.

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

```
1478  004262                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 20        *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
004262  000004                        TST20:  SCOPE
004264  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 2.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
004272  012737  000004  001124                MOV     #BIT2,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004320  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
004326  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004330  104002                                ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 2 FAILED TO BIT SET.

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

004332  000416                                BR      2$              ;/BR TO END SUBTEST.
004334                                  1$:                           ;/TRY CLEARING BIT 2.
004334  005037  001124                        CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004360  005737  001126                        TST     $BDDAT
004364  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

004366  104002                                ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 2 FAILED TO CLEAR.

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

```
 1533  004370                    2$:
```

```
                                        ;/*
                                        ;*********************************************************************
                                        ;*TEST 21        *TEST THAT CLOCK A STATUS REGISTER BIT 1 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
                                        ;*
                                        ;*
                                        ;*********************************************************************
004370  000004                  TST21:  SCOPE
004372  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE STATUS REGISTER.
                                                                ;/SET BIT 1.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                ;/READ THE STATUS REGISTER.
004400  012737  000002  001124          MOV     #BIT1,$GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004426  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
004434  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004436  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO bIT SET.

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

004440  000416                          BR      2$              ;/BR TO END SUBTEST.
004442                          1$:     ;/TRY CLEARING BIT 1.
004442  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004466  005737  001126                  TST     $BDDAT
004472  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

004474  104002                          ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO CLEAR.

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

```
  1588  004476                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 22        *TEST THAT CLOCK A STATUS REGISTER BIT 0 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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
004476  000004                          TST22:  SCOPE
004500  012737  000100  001166                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 0.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
004506  012737  000001  001124                  MOV     #BIT0,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004534  023737  001124  001126          1$:     CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
004542  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

004544  104002                                  ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER.
                                                                        ;/BIT 0 FAILED TO BIT SET.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

004546  000416                                  BR      2$              ;/BR TO eND SUBTEST.
004550                                  1$:                             ;/TRY CLEARING BIT 0.
004550  005037  001124                          CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
004574  005737  001126                          TST     $BDDAT
004600  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

004602  104002                                  ERROR   2               ;/ERROR-CLOCK A STATUS REGISTER.
                                                                        ;/BIT 0 FAILED TO cLEAR.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
 1643  004604                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 23       *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
004604  000004                        TST23:  SCOPE
004606  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 0.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
004614  012737  000001  001124                MOV     #BIT0,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
004642  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
004650  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004652  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 0 FAILED TO BIT SET.

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

004654  000416                        1$:     BR      2$              ;/BR TO END SUBTEST.
004656                                                                  ;/TRY CLEARING BIT 0.
004656  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
004702  005737  001126                        TST     $BDDAT
004706  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

004710  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 0 FAILED TO CLEAR.

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

```
1698  004712                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 24        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
004712  000004                        TST24:  SCOPE
004714  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 1.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
004722  012737  000002  001124                MOV     #BIT1,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
004750  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
004756  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

004760  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 1 FAILED TO BIT SET.

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

004762  000416                                BR      2$              ;/BR TO END SUBTEST.
004764                                  1$:                           ;/TRY CLEARING BIT 1.
004764  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005010  005737  001126                        TST     $BDDAT
005014  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

005016  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 1 FAILED TO CLEAR.

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

```
 1753  005020                    2$:
```

```
                                        ;/#
                                ;;**********************************************************************
                                ;*TEST 25        *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
                                ;*
                                ;*
                                ;;**********************************************************************
005020  000004                  TST25:  SCOPE
005022  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE BUFFER REGISTER.
                                                                ;/SET BIT 2.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE BUFFER REGISTER.
005030  012737  000004  001124          MOV     #BIT2,$GDDAT
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005056  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
005064  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005066  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 2 FAILED TO BIT SET.

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

005070  000416                          BR      2$              ;/BR TO END SUBTEST.
005072                          1$:                             ;/TRY CLEARING BIT 2.
005072  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005116  005737  001126                  TST     $BDDAT
005122  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

005124  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 2 FAILED TO CLEAR.

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

```
MAINDEC-11-DRLPG-A     MACY11 27(654)  15-DEC-77  08:29  PAGE 51
DRLPG.P11       T25    *TEST THAT CLOCK A BUFFER REGISTER BIT 2 CAN BE SET AND CLEARED                SEQ 0068

  1808  005126                        2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 26        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005126  000004                  TST26:  SCOPE
005130  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE BUFFER REGISTER.
                                                                ;/SET BIT 3.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE BUFFER REGISTER.
005136  012737  000010  001124          MOV     #BIT3,$GDDAT

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005164  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
005172  001402                         BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005174  104003                         ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO BIT SET.

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

005176  000416                         BR      2$              ;/BR TO END SUBTEST.
005200                          1$:             MOV     $GDDAT,@ABR     ;/TRY CLEARING BIT 3.
005200  005037  001124                 CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005224  005737  001126                 TST     $BDDAT
005230  001401                         BEQ     2$              ;/IF ZERO-NO ERROR!

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

005232  104003                         ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO CLEAR.

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

```
  1863  005234                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 27        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005234  000004                    TST27:  SCOPE
005236  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                  ;/CLEAR THE BUFFER REGISTER.
                                                                  ;/SET BIT 4.
                                                                  ;/SET FOR ERROR TYPEOUT S/B.
                                                                   ;/READ THE BUFFER REGISTER.
005244  012737  000020  001124            MOV     #BIT4,$GDDAT

                                ;*          MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*          MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005272  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 4 AND ONLY BIT 4 SET?
005300  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005302  104003                           ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                  ;/BIT 4 FAILED TO BIT SET.

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

005304  000416                           BR      2$              ;/BR TO END SUBTEST.
005306                          1$:             ;/TRY CLEARING BIT 4.
005306  005037  001124                   CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                  ;/NOW READ IT BACK.

                                ;*          MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*          MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005332  005737  001126                   TST     $BDDAT
005336  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

005340  104003                           ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                  ;/BIT 4 FAILED TO CLEAR.

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

```
  1918  005342                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 30        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005342  000004                        TST30:  SCOPE
005344  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 5.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
005352  012737  000040  001124                MOV     #BIT5,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005400  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
005406  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005410  104003                               ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 5 FAILED TO BIT SET.

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

005412  000416                               BR      2$              ;/BR TO END SUBTEST.
005414                          1$:                                   ;/TRY CLEARING BIT 5.
005414  005037  001124                       CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005440  005737  001126                       TST     $BDDAT
005444  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

005446  104003                               ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 5 FAILED TO CLEAR.

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

```
  1973  005450                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 31        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005450  000004                  TST31:  SCOPE
005452  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE BUFFER REGISTER.
                                                                ;/SET BIT 6.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE BUFFER REGISTER.
005460  012737  000100  001124          MOV     #BIT6,$GDDAT

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005506  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
005514  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005516  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 6 FAILED TO BIT SET.

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

005520  000416                          BR      2$              ;/BR TO END SUBTEST.
005522                          1$:     ;/TRY CLEARING BIT 6.
005522  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005546  005737  001126                  TST     $BDDAT
005552  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

005554  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 6 FAILED TO CLEAR.

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

```
2028  005556                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 32        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005556  000004                        TST32:  SCOPE
005560  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 7.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
005566  012737  000200  001124                MOV     #BIT7,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005614  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
005622  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005624  104003                            ERROR   3                   ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 7 FAILED TO BIT SET.

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

005626  000416                            BR      2$                  ;/BR TO END SUBTEST.
005630                                  1$:                           ;/TRY CLEARING BIT 7.
005630  005037  001124                    CLR     $GDDAT               ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005654  005737  001126                        TST     $BDDAT
005660  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

005662  104003                            ERROR   3                   ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 7 FAILED TO CLEAR.

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

```
  2083  005664                    2$:
```

```
                                        ;/#
                                        ;;**********************************************************************
                                        ;*TEST 33        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
005664  000004                        TST33:  SCOPE
005666  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 8.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
005674  012737  000400  001124                MOV     #BIT8,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005722  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 8 AND ONLY BIT 8 SET?
005730  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

005732  104003                                ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 8 FAILED TO bIT SET.

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

005734  000416                                BR      2$              ;/BR TO END SUBTEST.
005736                                  1$:                           ;/TRY CLEARING BIT 8.
005736  005037  001124                        CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
005762  005737  001126                        TST     $BDDAT
005766  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

005770  104003                                ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 8 FAILED TO CLEAR.

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

```
  2138  005772                    2$:
```

```
                                        ;/#
                                    ;;**********************************************************************
                                    ;*TEST 34        *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
                                    ;*
                                    ;*
                                    ;;**********************************************************************
005772  000004                    TST34:  SCOPE
005774  012737  000100  001166            MOV     #100,$TImES     ;;DO 100 ITERATIONS
                                                                  ;/CLEAR THE BUFFER REGISTER.
                                                                  ;/SET BIT 9.
                                                                  ;/SET FOR ERROR TYPEOUT S/B.
                                                                   ;/READ THE BUFFER REGISTER.
006002  012737  001000  001124            MOV     #BIT9,$GDDAT

                                    ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                    ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006030  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 9 AND ONLY BIT 9 SET?
006036  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006040  104003                           ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                  ;/BIT 9 FAILED TO BIT SET.

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

006042  000416                           BR      2$              ;/BR TO eND SUBTEST.
006044                          1$:                             ;/TRY CLEARING BIT 9.
006044  005037  001124                   CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                  ;/NOW READ IT BACK.

                                    ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                    ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006070  005737  001126                   TST     $BDDAT
006074  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

006076  104003                           ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                  ;/BIT 9 FAILED TO cLEAR.

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

```
  2193  006100                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 35       *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
006100  000004                        TST35:  SCOPE
006102  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 10.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
006110  012737  002000  001124                MOV     #BIT10,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006136  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 10 AND ONLY BIT 10 SET?
006144  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006146  104003                               ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 10 FAILED TO BIT SET.

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

006150  000416                               BR      2$              ;/BR TO END SUBTEST.
006152                          1$:                                   ;/TRY CLEARING BIT 10.
006152  005037  001124                       CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006176  005737  001126                       TST     $BDDAT
006202  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

006204  104003                               ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 10 FAILED TO CLEAR.

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

```
  2248  006206                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 36       *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
006206  000004                  TST36:  SCOPE
006210  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE BUFFER REGISTER.
                                                                ;/SET BIT 11.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE BUFFER REGISTER.
006216  012737  004000  001124          MOV     #BIT11,$GDDAT

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006244  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 11 AND ONLY BIT 11 SET?
006252  001402                         BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006254  104003                         ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 11 FAILED TO BIT SET.

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

006256  000416                         BR      2$              ;/BR TO END SUBTEST.
006260                          1$:                             ;/TRY CLEARING BIT 11.
006260  005037  001124                 CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006304  005737  001126                 TST     $BDDAT
006310  001401                         BEQ     2$              ;/IF ZERO-NO ERROR!

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

006312  104003                         ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 11 FAILED TO CLEAR.

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

```
  2303  006314                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 37       *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
006314  000004                  TST37:  SCOPE
006316  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE BUFFER REGISTER.
                                                                ;/SET BIT 12.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE BUFFER REGISTER.
006324  012737  010000  001124          MOV     #BIT12,$GDDAT

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006352  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 12 AND ONLY BIT 12 SET?
006360  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006362  104003                          ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 12 FAILED TO BIT SET.

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

006364  000416                          BR      2$              ;/BR TO END SUBTEST.
006366                          1$:     ;/TRY CLEARING BIT 12.
006366  005037  001124                  CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006412  005737  001126                  TST     $BDDAT
006416  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

006420  104003                          ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 12 FAILED TO CLEAR.

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

```
2358  006422                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 40        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
006422  000004                        TST40:  SCOPE
006424  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 13.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
006432  012737  020000  001124                MOV     #BIT13,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006460  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 13 AND ONLY BIT 13 SET?
006466  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006470  104003                               ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                        ;/BIT 13 FAILED TO bIT SET.

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

006472  000416                               BR      2$              ;/BR TO END SUBTEST.
006474                          1$:                                   ;/TRY CLEARING BIT 13.
006474  005037  001124                       CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006520  005737  001126                       TST     $BDDAT
006524  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

006526  104003                               ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                        ;/BIT 13 FAILED TO CLEAR.

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

```
  2413  006530                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 41        *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
006530  000004                        TST41:  SCOPE
006532  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 14.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
006540  012737  040000  001124                MOV     #BIT14,$GDDAT
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006566  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DID BIT 14 AND ONLY BIT 14 SET?
006574  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006576  104003                        ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 14 FAILED TO BIT SET.

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

006600  000416                        BR      2$              ;/BR TO END SUBTEST.
006602                          1$:                           ;/TRY CLEARING BIT 14.
006602  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006626  005737  001126                TST     $BDDAT
006632  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

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

006634  104003                        ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 14 FAILED TO cLEAR.

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

```
 2468  006636                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 42        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
006636  000004                    TST42:  SCOPE
006640  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE BUFFER REGISTER.
                                                                    ;/SET BIT 15.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE BUFFER REGISTER.
006646  012737  100000  001124            MOV     #BIT15,$GDDAT
                                ;*          MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*          MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006674  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 15 AND ONLY BIT 15 SET?
006702  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

006704  104003                           ERROR   3               ;/ERROR CLOCK AS BUFFER REGISTER.
                                                                ;/BIT 15 FAILED TO BIT SET.

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

006706  000416                           BR      2$              ;/BR TO END SUBTEST.
006710                          1$:                             ;/TRY CLEARING BIT 15.
006710  005037  001124                    CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*          MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*          MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
006734  005737  001126                    TST     $BDDAT
006740  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

006742  104003                           ERROR   3               ;/ERROR-CLOCK A BUFFER REGISTER.
                                                                ;/BIT 15 FAILED TO CLEAR.

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

```
2523  006744                    2$:
2524
2525
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 43        *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
006744  000004                        TST43:  SCOPE
006746  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 11.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
006754  012737  004000  001124                MOV     #BIT11,$GDDAT
                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007002  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 11 AND ONLY BIT 11 SET?
007010  001402                                BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007012  104005                                ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 11 FAILED TO BIT SET.

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

007014  000416                                BR      2$              ;/BR TO END SUBTEST.
007016                                  1$:                           ;/TRY CLEARING BIT 11.
007016  005037  001124                        CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007042  005737  001126                        TST     $BDDAT
007046  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

007050  104005                                ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 11 FAILED TO CLEAR.

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

```
2580  007052                    2$:
2581
```

```
                                        ;/#
                                        ;;**********************************************************************
                                        ;*TEST 44        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
007052  000004                        TST44:  SCOPE
007054  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 7.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
007062  012737  000200  001124                MOV     #BIT7,$GDDAT

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007110  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
007116  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007120  104005                               ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 7 FAILED TO BIT SET.

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


007122  000416                               BR      2$              ;/BR TO END SUBTEST.
007124                                  1$:                          ;/TRY CLEARING BIT 7.
007124  005037  001124                       CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007150  005737  001126                       TST     $BDDAT
007154  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

007156  104005                               ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 7 FAILED TO CLEAR.

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

```
2636  007160                    2$:
2637
```

```
                                        ;/*
                                        ;*********************************************************************
                                        ;*TEST 45       *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
                                        ;*
                                        ;*
                                        ;*********************************************************************
007160  000004                    TST45:  SCOPE
007162  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE STATUS REGISTER.
                                                                    ;/SET BIT 6.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE STATUS REGISTER.
007170  012737  000100  001124            MOV     #BIT6,$GDDAT

                                ;*          MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*          MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007216  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
007224  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007226  104005                           ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                 ;/BIT 6 FAILED TO BIT SET.

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

007230  000416                           BR      2$              ;/BR TO END SUBTEST.
007232                          1$:                             ;/TRY CLEARING BIT 6.
007232  005037  001124                    CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                 ;/NOW READ IT BACK.

                                ;*          MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*          MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007256  005737  001126                    TST     $BDDAT
007262  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

007264  104005                           ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                 ;/BIT 6 FAILED TO CLEAR.

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

```
2692  007266                    2$:
2693
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 46        *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
007266  000004                        $T46:   SCOPE
007270  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 5.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
007276  012737  000040  001124                MOV     #BIT5,$GDDAT

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007324  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
007332  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007334  104005                        ERROR   5                       ;/ERROR CLOCK BS STATUS REGISTER.
                                                                        ;/BIT 5 FAILED TO BIT SET.

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

007336  000416                               BR      2$              ;/BR TO END SUBTEST.
007340                          1$:                                   ;/TRY CLEARING BIT 5.
007340  005037  001124                       CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007364  005737  001126                       TST     $BDDAT
007370  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

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

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

```
 2748  007374                    2$:
 2749
```

```
                                        ;/#
                                ;;**************************************************************
                                ;*TEST 47        *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
                                ;*
                                ;*
                                ;;**************************************************************
007374  000004                TST47:  SCOPE
007376  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE STATUS REGISTER.
                                                                ;/SET BIT 4.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE STATUS REGISTER.
007404  012737  000020  001124          MOV     #BIT4,$GDDAT

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007432  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 4 AND ONLY BIT 4 SET?
007440  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007442  104005                        ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 4 FAILED TO BIT SET.

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

007444  000416                        BR      2$              ;/BR TO END SUBTEST.
007446                          1$:                           ;/TRY CLEARING BIT 4.
007446  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007472  005737  001126                TST     $BDDAT
007476  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

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

007500  104005                        ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 4 FAILED TO CLEAR.

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

```
 2804  007502                    2$:
 2805
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 50        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
007502  000004                    TST50:  SCOPE
007504  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE STATUS REGISTER.
                                                                    ;/SET BIT 3.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE STATUS REGISTER.
007512  012737  000010  001124            MOV     #BIT3,$GDDAT

                                ;*          MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*          MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007540  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
007546  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007550  104005                           ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 3 FAILED TO BIT SET.

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

007552  000416                           BR      2$              ;/BR TO END SUBTEST.
007554                          1$:                             ;/TRY CLEARING BIT 3,
007554  005037  001124                    CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*          MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*          MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007600  005737  001126                    TST     $BDDAT
007604  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

007606  104005                           ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 3 FAILED TO CLEAR.

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

```
2860  007610                    2$:
2861
```

```
                                        ;/*
                                        ;;*********************************************************************
                                        ;*TEST 51        *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
                                        ;*
                                        ;*
                                        ;;*********************************************************************
007610  000004                        TST51:  SCOPE
007612  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE STATUS REGISTER.
                                                                        ;/SET BIT 2.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE STATUS REGISTER.
007620  012737  000004  001124                MOV     #BIT2,$GDDAT
                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007646  023737  001124  001126        1$:     CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
007654  001402                        1$      BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007656  104005                        1$:     ERROR   5               ;/ERROR CLOCK B$ STATUS REGISTER.
                                                                        ;/BIT 2 FAILED TO BIT SET.

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

007660  000416                        1$:     BR      2$              ;/BR TO END SUBTEST.
007662                                                                  ;/TRY CLEARING BIT 2.
007662  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.
                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007706  005737  001126                        TST     $BDDAT
007712  001401                        1$      BEQ     2$              ;/IF ZERO-NO ERROR!

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

007714  104005                        1$:     ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                        ;/BIT 2 FAILED TO CLEAR.

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

```
2916  007716                    2$:
2917
```

```
                                        ;/#
                                ;;**********************************************************************
                                ;*TEST 52        *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
                                ;*
                                ;*
                                ;;**********************************************************************
007716  000004                TST52:  SCOPE
007720  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE STATUS REGISTER.
                                                                ;/SET BIT 1.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE STATUS REGISTER.
007726  012737  000002  001124          MOV     #BIT1,$GDDAT

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
007754  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
007762  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

007764  104005                        ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO BIT SET.

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

007766  000416                        BR      2$              ;/BR TO END SUBTEST.
007770                          1$:                           ;/TRY CLEARING BIT 1.
007770  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
010014  005737  001126                TST     $BDDAT
010020  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

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

010022  104005                        ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO CLEAR.

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

```
2972  010024                    2$:
2973
```

```
                                        ;/*
                                        ;*********************************************************************
                                        ;*TEST 53        *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
                                        ;*
                                        ;*
                                        ;*********************************************************************
010024  000004                  TST53:  SCOPE
010026  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE STATUS REGISTER.
                                                                ;/SET BIT 0.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE STATUS REGISTER.
010034  012737  000001  001124          MOV     #BIT0,$GDDAT

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
010062  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
010070  001402                         BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

010072  104005                         ERROR   5               ;/ERROR CLOCK BS STATUS REGISTER.
                                                                ;/BIT 0 FAILED TO BIT SET.

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

010074  000416                         BR      2$              ;/BR TO END SUBTEST.
010076                          1$:             ;/TRY CLEARING BIT 0.
010076  005037  001124                 CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
010122  005737  001126                 TST     $BDDAT
010126  001401                         BEQ     2$              ;/IF ZERO-NO ERROR!

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

010130  104005                         ERROR   5               ;/ERROR-CLOCK B STATUS REGISTER.
                                                                ;/BIT 0 FAILED TO CLEAR.

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

```
  3028  010132                    2$:
```

```
3029                                            ;/*
3030                                            ;;*****************************************************************
3031                                            ;*TEST 54        *TEST THAT CLOCK B BUFFER REGISTER BIT 0 CAN BE SET AND CLEARED
3032                                            ;*
3033                                            ;*CLOCK B BUFFER REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
3034                                            ;*F/FS OR GATES
3035                                            ;*
3036                                            ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
3037                                            ;*
3038                                            ;*
3039                                            ;;*****************************************************************
3040  010132  000004                            TST54:  SCOPE
3041  010134  012737  000100  001166                    MOV     #100,$TIMES     ;;DO 100 ITERATIONS
3042                                                                            ;/CLEAR THE BUFFER REGISTER.
3043                                                                            ;/SET BIT 0.
3044                                                                            ;/SET FOR ERROR TYPEOUT S/B.
3045                                                                             ;/READ THE BUFFER REGISTER.
3046  010142  012737  000001  001124                    MOV     #BIT0,$GDDAT
3047
3048                                          ;*        MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
3049
3050                                          ;*        MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
3051  010170  023737  001124  001126                    CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
3052  010176  001402                                   BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
3053
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

3057  010200  104006                                  ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
3058                                                                            ;/BIT 0 FAILED TO BIT SET.
3059
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

3063  010202  000416                                  BR      2$              ;/BR TO END SUBTEST.
3064  010204                                  1$:                             ;/TRY CLEARING BIT 0.
3065  010204  005037  001124                          CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
3066                                                                            ;/NOW READ IT BACK.
3067
3068                                          ;*        MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
3069
3070                                          ;*        MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
3071  010230  005737  001126                          TST     $BDDAT
3072  010234  001401                                  BEQ     2$              ;/IF ZERO-NO ERROR!
3073
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

3077  010236  104006                                  ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
3078                                                                            ;/BIT 0 FAILED TO CLEAR.
3079
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
  3083  010240                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 55        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
010240  000004                        TST55:  SCOPE
010242  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 1.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
010250  012737  000002  001124                MOV     #BIT1,$GDDAT

                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                                        ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010276  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
010304  001402                               BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

010306  104006                               ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                        ;/BIT 1 FAILED TO BIT SET.

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

010310  000416                               BR      2$              ;/BR TO END SUBTEST.
010312                                  1$:                          ;/TRY CLEARING BIT 1.
010312  005037  001124                       CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.

                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                                        ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010336  005737  001126                       TST     $BDDAT
010342  001401                               BEQ     2$              ;/IF ZERO-NO ERROR!

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

010344  104006                               ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                        ;/BIT 1 FAILED TO CLEAR.

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

```
  3138  010346                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 56        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
010346  000004                          TST56:  SCOPE
010350  012737  000100  001166                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 2.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
010356  012737  000004  001124                  MOV     #BIT2,$GDDAT
                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                        ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010404  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
010412  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

010414  104006                          ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 2 FAILED TO BIT SET.

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

010416  000416                          BR      2$              ;/BR TO END SUBTEST.
010420                          1$:                             ;/TRY CLEARING BIT 2.
010420  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                        ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010444  005737  001126                  TST     $BDDAT
010450  001401                          BEQ     2$              ;/IF ZERO-NO ERROR!

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

010452  104006                          ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 2 FAILED TO CLEAR.

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

```
 3193  010454                    2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 57        *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
                                ;*
                                ;*
                                ;;**********************************************************************
010454  000004                 TST57:  SCOPE
010456  012737  000100  001166         MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                ;/CLEAR THE BUFFER REGISTER.
                                                                ;/SET BIT 3.
                                                                ;/SET FOR ERROR TYPEOUT S/B.
                                                                 ;/READ THE BUFFER REGISTER.
010464  012737  000010  001124         MOV     #BIT3,$GDDAT

                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010512  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;/DID BIT 3 AND ONLY BIT 3 SET?
010520  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

010522  104006                         ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO BIT SET.

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

010524  000416                         BR      2$              ;/BR TO END SUBTEST.
010526                          1$:                             ;/TRY CLEARING BIT 3.
010526  005037  001124                 CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.

                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                                ;*      MOV     @BBR,$BDDAT     ;/READ DEViCE REG BBR,PUT DATA IN $BDDAT.
010552  005737  001126                 TST     $BDDAT
010556  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

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

010560  104006                         ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 3 FAILED TO CLEAR.

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

```
  3248  010562                    2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 60        *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
                                        ;*
                                        ;*
                                        ;;**********************************************************************
010562  000004                        TST60:  SCOPE
010564  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 4.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
010572  012737  000020  001124                MOV     #BIT4,$GDDAT
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010620  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DID BIT 4 AND ONLY BIT 4 SET?
010626  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

010630  104006                        ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 4 FAILED TO BIT SET.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

010632  000416                        BR      2$              ;/BR TO END SUBTEST.
010634                          1$:                             ;/TRY CLEARING BIT 4.
010634  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010660  005737  001126                TST     $BDDAT
010664  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

010666  104006                        ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 4 FAILED TO CLEAR.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
 3303  010670                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 61        *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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
010670  000004                        TST61:  SCOPE
010672  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 5.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
010700  012737  000040  001124                MOV     #BIT5,$GDDAT
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010726  023737  001124  001126        1$:     CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
010734  001402                        1$      BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

010736  104006                                ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                        ;/BIT 5 FAILED TO BIT SET.

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

010740  000416                                BR      2$              ;/BR TO END SUBTEST.
010742                                  1$:                           ;/TRY CLEARING BIT 5.
010742  005037  001124                        CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                        ;/NOW READ IT BACK.
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
010766  005737  001126                        TST     $BDDAT
010772  001401                                BEQ     2$              ;/IF ZERO-NO ERROR!

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

010774  104006                                ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                        ;/BIT 5 FAILED TO CLEAR.

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

```
  3358  010776                    2$:
```

```
                                        ;/*
                                        ;;*********************************************************************
                                        ;*TEST 62       *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
                                        ;*
                                        ;*
                                        ;;*********************************************************************
010776  000004                    TST62:  SCOPE
011000  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                    ;/CLEAR THE BUFFER REGISTER.
                                                                    ;/SET BIT 6.
                                                                    ;/SET FOR ERROR TYPEOUT S/B.
                                                                     ;/READ THE BUFFER REGISTER.
011006  012737  000100  001124            MOV     #BIT6,$GDDAT

                                ;*          MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                                ;*          MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
011034  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
011042  001402                           BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

011044  104006                           ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                 ;/BIT 6 FAILED TO BIT SET.

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

011046  000416                           BR      2$              ;/BR TO END SUBTEST.
011050                          1$:                             ;/TRY CLEARING BIT 6.
011050  005037  001124                   CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                 ;/NOW READ IT BACK.

                                ;*          MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR

                                ;*          MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
011074  005737  001126                   TST     $BDDAT
011100  001401                           BEQ     2$              ;/IF ZERO-NO ERROR!

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

011102  104006                           ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                 ;/BIT 6 FAILED TO CLEAR.

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

```
  3413  011104                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 63        *TEST THAT CLOCK B BUFFER REGISTER BIT 7 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
                                        ;*
                                        ;*
                                        ;;*****************************************************************
011104  000004                        TST63:  SCOPE
011106  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                                        ;/CLEAR THE BUFFER REGISTER.
                                                                        ;/SET BIT 7.
                                                                        ;/SET FOR ERROR TYPEOUT S/B.
                                                                         ;/READ THE BUFFER REGISTER.
011114  012737  000200  001124                MOV     #BIT7,$GDDAT
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
011142  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DID BIT 7 AND ONLY BIT 7 SET?
011150  001402                        BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

011152  104006                        ERROR   6               ;/ERROR CLOCK BS BUFFER REGISTER.
                                                                ;/BIT 7 FAILED TO BIT SET.

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

011154  000416                        BR      2$              ;/BR TO END SUBTEST.
011156                          1$:                           ;/TRY CLEARING BIT 7.
011156  005037  001124                CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
                                                                ;/NOW READ IT BACK.
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                ;*      MOV     @BBR,$BDDAT     ;/READ DEVICE REG BBR,PUT DATA IN $BDDAT.
011202  005737  001126                TST     $BDDAT
011206  001401                        BEQ     2$              ;/IF ZERO-NO ERROR!

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

011210  104006                        ERROR   6               ;/ERROR-CLOCK B BUFFER REGISTER.
                                                                ;/BIT 7 FAILED TO CLEAR.

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

```
011212                          2$:
                                .SBTTL  *
                                .SBTTL  * PHASE 2 ADVANCED BASIC LOGIC TESTS
                                .SBTTL  *


                                ;;**********************************************************************
                                ;*TEST 64        *TEST THAT CLOCK A'S COUNT REGISTER IS CLEAR

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

                                ;;**********************************************************************
011212  000004                  TST64:  SCOPE
011214  012737  000002  001166          MOV     #2,$TIMES       ;;DO 2 ITERATIONS

                                                                ;SELECT MODE 0.
                                                                ;CLEAR THE BUFFER REGISTER. BUFFER
                                                                ;REGISTER WILL BE TRANSFERRED TO
                                                                ;COUNT REGISTER SINCE THIS IS MODE 0.
011222  005037  001124                  CLR     $GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
011246  005037  001124                  CLR     $GDDAT          ;EXPECT TO READ BACK ALL 0'S.
                                                                ;READ THE COUNT REGISTER - IT SHOULD BE CLEAR.
                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
011262  005737  001126                  TST     $BDDAT
011266  001401                          BEQ     1$              ;BR IF YES TO NEXT TEST

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

011270  104004                  ERROR   4               ;ERROR - CLOCK A'S COUNTER REGISTER
                                                        ;NOT CLEAR.

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

011272                          1$:

                                ;;**********************************************************************
                                ;*TEST 65        *TEST CLOCK A'S COUNT REGISTER WITH 125252 PATTERN
```

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

                                        ;;*****************************************************************
011272  000004                          TST65:  SCOPE
011274  012737  000100  001166                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS

                                                                        ;SELECT MODE 0.
                                                                        ;LOAD THE BUFFER REGISTER WITH
                                                                        ;PATTERN 125252. IT WILL BE
                                                                        ;TRANSFERRED TO THE COUNT REGISTER
                                                                        ;SINCE THIS IS MODE 0.
011302  005037  001124                  CLR     $GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

011316  012737  125252  001124          MOV     #125252,$GDDAT  ;SET EXPECTED TO PATTERN IN CASE OF
                                                                ;NEED OF ERROR TYPEOUT.
                                                                ;READ THE COUNT REGISTER

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR

                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.

011344  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
                                                                ;COME THROUGH?
011352  001401                          BEQ     1$              ;BR IF YES TO NEXT TEST.

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

011354  104004                          ERROR   4               ;DATA ERROR CLOCK A + PATTERN "125252"
                                                                ;FAILED TO TRANSFER PROPERLY BETWEEN
                                                                ;BUFFER AND COUNT REGISTERS.

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

011356                          1$:

                                        ;;*****************************************************************
                                        ;*TEST 66        *TEST CLOCK A'S COUNT REGISTER WITH 052525 PATTERN

                                        ;*
                                        ;*CLOCK A COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
```

```
3576                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
3577                                    ;*
3578                                    ;*                                  ;
3579
3580                                    ;;*****************************************************************
3581  011356  000004                    TST66:  SCOPE
3582  011360  012737  000050  001166            MOV     #50,$TIMES      ;;DO 50 ITERATIONS
3583
3584                                                                    ;SELECT MODE 0.
3585  011366  005037  001124            CLR     $GDDAT
3586
3587                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3588                                                                    ;LOAD THE BUFFER REGISTER WITH
3589                                                                    ;PATTERN 052525. IT WILL BE
3590                                                                    ;TRANSFERRED TO THE COUNT REGISTER
3591                                                                    ;SINCE THIS IS MODE 0.
3592
3593  011402  012737  052525  001124      MOV     #052525,$GDDAT  ;SET EXPECTED TO PATTERN IN CASE OF
3594                                                                    ;NEED OF ERROR TYPEOUT.
3595                                                                    ;READ THE COUNT REGISTER
3596
3597                                    ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
3598
3599                                    ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
3600
3601  011430  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
3602                                                                    ;COME THROUGH?
3603  011436  001401                    BEQ     1$              ;BR IF YES TO nEXT TEST.
3604
3605
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

3609  011440  104004                            ERROR   4               ;DATA ERROR CLOCK A ← PATTERN "052525"
3610                                                                    ;FAILED TO TRANSFER PROPERLY BETWEEN
3611                                                                    ;BUFFER AND COUNT REGISTERS.
3612
3613
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

3617  011442                            1$:
3618
3619
3620                                    ;;*****************************************************************
3621                                    ;*TEST 67        *TEST THAT CLOCK B'S COUNT REGISTER IS CLEAR
3622
3623                                    ;*
3624                                    ;*CLOCK B COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
3625                                    ;*F/FS OR GATES
3626                                    ;*
3627                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
3628                                    ;*
3629                                    ;*
```

```
                        ;;**********************************************************************
011442  000004         TST67:  SCOPE
011444  012737  000002  001166        MOV     #2,$TIMES       ;;DO 2 ITERATIONS
                                                              ;SELECT MODE 0.
                                                              ;CLEAR THE BUFFER REGISTER. BUFFER
                                                              ;REGISTER WILL BE TRANSFERRED TO
                                                              ;COUNT REGISTER SINCE THIS IS MODE 0.
011452  005037  001124                CLR     $GDDAT
                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
011476  005037  001124                CLR     $GDDAT          ;EXPECT TO rEAD BACK ALL 0'S.
                                                              ;READ THE COUNT REGISTER - IT SHOULD BE CLEAR.
                                ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
011512  005737  001126                TST     $BDDAT
011516  001401                        BEQ     1$              ;BR IF YES TO NEXT TESt

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

011520  104007                        ERROR   7               ;ERROR - CLOCK B'S COUNTER REGISTER
                                                              ;NOT CLEAR.

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


011522                          1$:

                        ;;**********************************************************************
                        ;*TEST 70        *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
                        ;*
                        ;*
                        ;;**********************************************************************
011522  000004         TST70:  SCOPE
011524  012737  000100  001166        MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                                              ;SELECT MODE 0.
011532  005037  001124                CLR     $GDDAT
```

```
3684                                    ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
3685                                                                    ;LOAD THE BUFFER REGISTER WITH
3686                                                                    ;PATTERN 125. IT WILL BE
3687                                                                    ;TRANSFERRED TO THE COUNT REGISTER
3688                                                                    ;SINCE THIS IS MODE 0.
3689
3690  011546  012737  000125  001124          MOV     #125,$GDDAT     ;SET EXPECTED TO PATTERN IN CASE OF
3691
3692                                    ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
3693                                                                    ;NEED OF ERROR TYPEOUT.
3694                                                                    ;READ THE COUNT REGISTER
3695
3696                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
3697
3698  011574  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
3699                                                                    ;COME THROUGH?
3700  011602  001401                          BEQ     1$              ;BR IF YES TO NEXT TEST.
3701
3702
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3706  011604  104007                          ERROR   7               ;DATA ERROR CLOCK B - PATTERN "125"
3707                                                                    ;FAILED TO TRANSFER PROPERLY BETWEEN
3708                                                                    ;BUFFER AND COUNT REGISTERS.
3709
3710
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3714  011606                            1$:

3715
3716                                    ;;**************************************************************
3717                                    ;*TEST 71        *TEST CLOCK B'S COUNT REGISTER WITH 252 PATTERN
3718
3719                                    ;*
3720                                    ;*CLOCK B COUNT REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
3721                                    ;*F/FS OR GATES
3722                                    ;*
3723                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 OCCURANCES PER PASS
3724                                    ;*
3725                                    ;*
3726
3727                                    ;;**************************************************************
3728  011606  000004                    TST71:  SCOPE
3729  011610  012737  000100  001166            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
3730
3731                                                                    ;SELECT MODE 0.
3732  011616  005037  001124                    CLR     $GDDAT
3733
3734                                    ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
3735                                                                    ;LOAD THE BUFFER REGISTER WITH
3736                                                                    ;PATTERN 252. IT WILL BE
3737                                                                    ;TRANSFERRED TO THE COUNT REGISTER
```

```
3738                                                            ;SINCE THIS IS MODE 0.
3739
3740  011632  012737  000252  001124          MOV     #252,$GDDAT     ;SET EXPECTED TO PATTERN IN CASE OF
3741
3742                                  ;*        MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
3743                                                                    ;NEED OF ERROR TYPEOUT.
3744                                                                    ;READ THE COUNT REGISTER
3745
3746                                  ;*        MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
3747
3748  011660  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
3749                                                                    ;COME THROUGH?
3750  011666  001401                          BEQ     1$              ;BR IF YES TO NEXT TEST.
3751
3752
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3756  011670  104007                          ERROR   7               ;DATA ERROR CLOCK B - PATTERN "252"
3757                                                                    ;FAILED TO TRANSFER PROPERLY BETWEEN
3758                                                                    ;BUFFER AND COUNT REGISTERS.
3759
3760
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3764  011672                          1$:

3766
3767                                  ;;*****************************************************************
3768                                  ;*TEST 72       *TEST THE SETTING OF MAINTENANCE STP1 IN CLOCK A BIT 15 TO SET
3769                                  ;*
3770                                  ;*NEW SIGNALS   GENERATION OF "STP1" BY "LD STAT A" H + "BD 12" H
3771                                  ;*
3772                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
3773                                  ;*
3774                                  ;*
3775                                  ;;*****************************************************************
3776  011672  000004                  TST72:  SCOPE
3777
3778  011674  005037  001124                  CLR     $GDDAT
3779
3780                                  ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3781
3782                                  ;*        MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
3783                                                                    ;MAKE SURE THE STATUS REGISTER IS CLEAR.
3784                                                                    ;SET MAINTENANCE STP1.
3785  011720  012737  010000  001124          MOV     #BIT12,$GDDAT
3786
3787                                  ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3788
3789                                                                    ;DID BIT15 (STP1 FLAG) SET?
3790
3791                                  ;*        MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
```

```
3792  011746  005737  001126                  TST     $BDDAT
3793  011752  100401                          BMI     1$              ;BR IF YES - NEXT TEST
3794
3795
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3799  011754  104001                          ERROR   1               ;ERROR - MAINTENANCE STP1 (BIT12)
3800                                                                  ;DID NOT SET BIT15 (STP1 FLAG) CLOCK A.
3801
3802
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$


3806  011756                        1$:
3807
3808
3809                                ;;*****************************************************************
3810                                ;*TEST 73        *TEST THAT BIT00 IN CLOCK A STATUS REG. WILL SET WHEN BIT13 AND MAIN. ST
3811                                ;*
3812                                ;*NEW SIGNALS    "STP1" H + "ST1 ENB CNTR (1)" H DIRECT SETTING
3813                                ;*              "ST1 FLAG"
3814                                ;*
3815                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "DEVICE OUT" 2 PER PASS
3816                                ;*
3817                                ;*
3818                                ;*****************************************************************
3819  011756  000004                TST73:  SCOPE
3820
3821                                                                  ;SET "ST1 ENB COUNTER" IN CLK A'S STATUS REG.
3822                                                                  ;GENERATE A MAINTENANCE ST1.
3823                                                                  ;DID BIT00 (ENABL CNTR A) SET?
3824  011760  012737  020000  001124        MOV     #BIT13,$GDDAT
3825
3826                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3827
3828                                ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
3829  012006  052737  010000  001124        BIS     #BIT12,$GDDAT
3830
3831                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3832
3833                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
3834  012034  032737  000001  001126        BIT     #BIT00,$BDDAT
3835  012042  001001                        BNE     1$              ;BR IF YES - NEXT TEST.
3836
3837
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3841  012044  104001                          ERROR   1               ;ERROR - BIT00 OF CLOCK A'S STATUS REGISTER
3842                                                                  ;FAILED TO SET WHEN BIT13 WAS SET
3843                                                                  ;AND A MAINTENANCE ST1 GENERATED.
3844
3845
```

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

3849  012046                          1$:                             ;LEAVE SUBTEST WITH CLOCK CLEAR.
3850  012046  005037  001124                  CLR     $GDDAT
3851
3852                                  ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3853
3854
3855                                  ;;*****************************************************************
3856                                  ;*TEST 74        *TEST THAT CLOCK A WILL INCREMENT - MODE 0 - RATE STP1 FIRST COUNT TEST
3857                                  ;*
3858                                  ;*COUNT TEST - THIS IS THE VERY FIST TIME THAT THE COUNTER
3859                                  ;*HAS BEEN ASKED TO INCREMENT! WHAT WE ARE GOING TO DO IS
3860                                  ;*CLEAR THE BUFFER, SELECT MODE 0, RATE OF STP1.
3861                                  ;*NEXT WILL GENERATE THE STP1 THROUGH MAINTENANCE MODE (WE'VE
3862                                  ;*DONE THIS BEFORE IN A PREVIOUS TEST TO SEE IF THE FLAG WOULD SET)
3863                                  ;*AND SEE IF THE COUNTER HAS INCREMENTED.
3864                                  ;*
3865                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
3866                                  ;*
3867                                  ;*
3868                                  ;*****************************************************************
3869  012062  000004                  TST74:  SCOPE
3870
3871                                                                  ;CLEAR CLOCK A.
3872                                                                  ;CLEAR BUFFER REGISTER.
3873  012064  005037  001124                  CLR     $GDDAT
3874
3875                                  ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3876
3877                                  ;*        MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
3878                                                                  ;SELECT: MODE0! RATE "STP1" AND ENABLE
3879                                                                  ;CLOCK A TO COUNT.
3880                                                                  ;NOW GENERATE A MAINTENANCE STP1 - AT
3881                                                                  ;THIS TIME THE CLOCK SHOULD COUNT ONCE.
3882  012110  012737  000015  001124          MOV     #15,$GDDAT
3883
3884                                  ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3885
3886                                  ;*        MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
3887  012136  052737  010000  001124          BIS     #BIT12,$GDDAT
3888
3889                                  ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
3890  012154  012737  000001  001124          MOV     #1,$GDDAT       ;FOR ERROR TYPEOUT (IF NEEDED) SET THE #1.
3891                                                                  ;READ THE COUNT REGISTER - SHOULD=1.
3892
3893                                  ;*        MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
3894  012172  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THE COUNTER COUNT ONCE?
3895  012200  001401                          BEQ     1$              ;IF YES - BR NEXT TEST.
3896
3897
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
3901  012202  104011                          ERROR   11              ;ERROR - COUNT A FAILED TO COUNT
3902                                                                  ;ONCE, WHEN ENABLED, MODE 0
3903                                                                  ;RATE "STP1" SEE ABOVE COMMENTS
3904                                                                  ;FOR COMPLETE DEScRIPTION AND LIST
3905                                                                  ;OF EVENTS.
3906
3907
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3911  012204                          1$:
3912
3913
3914                                  ;;*****************************************************************
3915                                  ;*TEST 75       *TEST THE ABILITY OF CLOCK A TO COUNT FROM ZERO TO OVERFLOW USING M STP1
3916                                  ;*
3917                                  ;*IN THIS TES WE'LL COUNT THE COUNTER THROUGH EACH STEP
3918                                  ;*FROM ZERO TO OVERFLOW USING MAINTENANCE "STP1" COUNTS.
3919                                  ;*IT IS KNOWN THAT THE COUNTER WILL INCREMENT ONCE USING
3920                                  ;*THE MAINTENANCE STP1 AND THAT THE COUNTER F/FS WILL
3921                                  ;*PASS ALL DATA BITS.
3922                                  ;*UNKNOW IS THE ABILITY OF THE F/F'S TO PROPAGATE THEIR
3923                                  ;*OVERFLOWS.
3924                                  ;*
3925                                  ;*IF IT IS DESIRED TO START THIS TEST AT A VALUE OTHER THAN
3926                                  ;*ZERO, CHANGE THE SECOND INSTR. OF THIS TEST TO A VALUE TO BE
3927                                  ;*LOADED INTO THE BUFFER TO THAT VALUE DESIRED.
3928                                  ;*
3929                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
3930                                  ;*
3931                                  ;*
3932                                  ;;*****************************************************************
3933  012204  000004                  TST75:  SCOPE
3934  012206  012737  000001  001166          MOV     #1,$TIMES       ;;DO 1 ITERATION
3935
3936  012214  005037  001124                  CLR     $GDDAT  ;START THE COUNTER FROM ZERO.
3937                                                                  ;NOTE: A VALUE OTHER THAN ZERO MAY BE
3938                                                                  ;PATCHED IN HERE IN ORDER THAT A COUNT
3939                                                                  ;MAY BE STARTED HIGHER.
3940  012220                          1$:                             ;DISABLE CLOCK A.
3941  012220  005037  001356                  CLR     $TMDAT
3942
3943                                  ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
3944
3945                                  ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
3946                                                                  ;LOAD THE COUNTER BUFFER WITH VALUE.
3947                                                                  ;"1$" IS THE LOOP BACK POINT ON
3948                                                                  ;"LOOP ON TEST" (SW14=1) FEATURE. NOWMAL
3949                                                                  ;LOOP BACK POINT WILL BE "2$".
3950
3951  012244  012737  000015  001356          MOV     #15,$TMDAT
3952
3953                                  ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
```

```
                                                                ;ENABLE COUNTER TO COUNT. SELECTED:
                                                                ;MODE 0, RATE "STP1"
012262                          2$:
                        ;*      MOV     @ASR,$TMDAT             ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
012272  052737  010000  001356          BIS     #BIT12,$TMDAT   ;GENERATE A MAINTENANCE "STP1". THIS
                        ;*      MOV     $TMDAT,@ASR             ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
                                                                ;ONE VALUE.
012310  005237  001124          INC     $GDDAT                  ;$GDDAT IS USED TO KEEP TRACK OF THE
                                                                ;VALUE THE CLOCK SHOULD COUNT TO.
                        ;*      MOV     @ACR,$BDDAT             ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
012324  023737  001126  001124  CMP     $BDDAT,$GDDAT           ;DID COUNT OCCUR CORRECTLY?
012332  001402          BEQ     3$                              ;IF YES - BR TO "3$".

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

012334  104011          ERROR   11                              ;ERROR - CLOCK A FAILED TO UPCOUNT
                                                                ;CORRECTLY USING MODE 0 - RATE "STP1".

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

012336  000403          BR      4$
012340  005737  001124  3$:     TST     $GDDAT                  ;DID WE ACHIEVE OVERFLOW TO ZERO YET?
012344  001410          BEQ     5$                              ;IF YES - BR NEXT TEST

012346  032777  040000  166564  4$:     BIT     #BIT14,@SWR     ;LOOP CURRENT COUNT?
012354  001742          BEQ     2$                              ;BR IF NO TO nEXT COUNT UPDATE.
012356  162737  000001  001124          SUB     #1,$GDDAT       ;IF YES - DECREMENT EXPECTED AND RELOAD.
012364  000715          BR      1$                              ;GOTO RELOAD POINT.

012366                          5$:                             ;END SUBTEST

                        .SBTTL  *
                        .SBTTL  * PHASE 3 CLOCK A COUNT FUNCTION TESTS
                        .SBTTL  *


                        ;;**********************************************************************
                        ;*TEST 76        *TEST THAT CLOCK A OVERFLOW WILL OCCUR
                        ;*
                        ;*NOW WE'LL TRY AND GENERATE "A OVERFLOW L"
                        ;*WHICH SETS BDO5. WE'LL DO IT BY PRESETTING THE BUFFER
                        ;*AT 177777 AND GENERATE A MAINTENANCE STP1. WE ALREADY
                        ;*KNOW MAINTENANCE STP1'S WILL ADVANCE THE COUNTER.
                        ;*ALSO SEE IF "MODE" FLG GETS SET.
                        ;*
```

```
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************************
012366  000004                        TST76:  SCOPE

                                                                ;MAKE SURE CLOCK A CLEAR.
012370  005037  001124                        CLR     $GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
012404  012737  177777  001124         ;*      MOV     #177777,$GDDAT  ;PRESET BUFFER TO ALL ONES.
                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                                                        ;SELECT MODE 0; RATE STP4; GO.
                                                                        ;GENERATE A MAINTENANCE STP1.
012422  012737  000015  001124                MOV     #15,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
012450  052737  010000  001124                BIS     #BIT12,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
012466                                  1$:                             ;DID OVERFLOW BIT SET?
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
012476  032737  000040  001126                BIT     #BIT05,$BDDAT
012504  001002                                BNE     2$              ;BR IF YES TO "2$".

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

012506  104012                                ERROR   12              ;ERROR BIT05 OF CSR CLOCK A FAILED
                                                                        ;TO SET ON OVERFLOW.

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

012510  000411                                BR      3$

012512                                  2$:                             ;DID BIT07 - "MODE" FLG SET?

                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
012522  032737  000200  001126                BIT     #BIT07,$BDDAT
                                                                        ;IT SHOULD SET WHEN BIT05 SETS.
012530  001001                                BNE     3$              ;BR IF YES TO 3$.

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

012532  104012                                ERROR   12              ;OVERFLOW FAILED TO SET CLOCK A'S
                                                                        ;CSR BIT07 "MODE" FLG. IT
                                                                        ;HAD; HOWEVER SET BIT05 "OVERFLOW"
                                                                        ;FLAG.
```

```
4062
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4066  012534                                3$:
4067                                        ;;*****************************************************************
4068                                        ;*TEST 77       *TEST IN CLOCK A THAT OVERFLOW IN MODE 0 CAUSE CLEARING OF "ENB CNTR" F/
4069                                        ;*
4070                                        ;*WE'RE GOING TO SEE IF "A RELOAD" H AND "MODE 0" H CLEAR
4071                                        ;*"ENB CNTR A" F/F.  "A RELOAD" GENERATED ON OVERFLOW AND
4072                                        ;*"MODE 0" GENERATED BY "MODE A0 (0)" AND "MODE A0 (1)"
4073                                        ;*ALSO SEE IF COUNTER REGISTER GETS RELOADED
4074                                        ;*
4075                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
4076                                        ;*
4077                                        ;*
4078                                        ;*****************************************************************
4079  012534  000004                        TST77:  SCOPE
4080
4081                                                                        ;MAKE SURE CLOCK A IS CLEAR.
4082  012536  005037  001124                        CLR     $GDDAT
4083
4084                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4085                                                                        ;PRESET BUFFER TO ALL ONES.
4086                                                                        ;SELECT MODE 0, RATE "STP4"; GO.
4087                                                                        ;GENERATE A MAINTENANCE STPI.
4088                                                                        ;THIS SHOULD CAUSE AN OVERFLOW
4089                                                                        ;OVERFLOW WILL GENERATE "A RELOAD" H
4090                                                                        ;COMBINE THIS WITH MODE 0 AND WE
4091  012552  012737  177777  001124                MOV     #177777,$GDDAT
4092
4093                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
4094  012570  012737  000015  001124                MOV     #15,$GDDAT
4095
4096                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4097
4098                                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
4099  012616  052737  010000  001124                BIS     #BIT12,$GDDAT
4100
4101                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4102                                                                        ;SHOULD CLEAR "ENB CNTR A" F/F.
4103
4104
4105                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
4106  012644  032737  000001  001356                BIT     #BIT00,$TMDAT   ;DID BIT00 "ENB CNTR A" F/F CLEAR?
4107  012652  001402                               BEQ     1$              ;BR IF YES - NEXT TEST.
4108
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$


4112  012654  104012                            ERROR   12                  ;"ENB CNTR A" F/F NOT CLEARED
4113                                                                        ;ON OVERFLOW.
4114
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
4118  012656  000411                          BR      2$
4119
4120  012660                          1$:
4121
4122                                  ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
4123  012670  022737  177777  001126          CMP     #177777,$BDDAT  ;DID COUNTER GET RELOADED AFTER
4124                                                                  ;THE OVERFLOW OCCURRED?
4125  012676  001401                          BEQ     2$              ;BR IF YES - NEXT TEST.
4126
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4130  012700  104012                          ERROR   12              ;ERROR CLOCK A - COUNTER DID NOT GET RELOAD FROM
4131                                                                  ;BUFFER AFTER OVERFLOW. "A RELOAD" H
4132                                                                  ;DID GET GENERATED ON WE WOULD
4133                                                                  ;HAVE GOT PREVIOUS ERROR; THEREFORE PROBLEM
4134                                                                  ;MUST EXIST WITH COMBINING "A RELOAD" H
4135                                                                  ;WITH "MODE A1 (0)" H TO GENERATE A
4136                                                                  ;COUNTER LOAD.
4137
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4141  012702                          2$:
4142
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 100      *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"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
012702  000004                        TST100: SCOPE
012704  012737  000020  001166              MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                    ;/MAKE SURE CLOCK IS CLEAR.
                                                                    ;/CLEAR THE BUFFER.
                                                                    ;/SELECT: MODE 0, RATE 1MHZ ; GO.
012712  005037  001124                    CLR     $GDDAT

                                        ;*  MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*  MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
012736  012737  000003  001124          MOV     #1!2,$GDDAT

                                        ;*  MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
012754  005000                          CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
012756  005200                  1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
012760  001376                          BNE     1$

                                                                    ;/DID COUNTER INCREMENT AT ALL?

                                        ;*  MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
012772  005737  001126                  TST     $BDDAT
012776  001011                          BNE     2$              ;/IF YES - BR NEXT TEST.

                                                                    ;/COUNTER MAY HAVE HAD TIME TO oVERFLOW.

                                        ;*  MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
013010  032737  000040  001126          BIT     #BIT05,$BDDAT
                                                                    ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                    ;/F/F HAD SET.
013016  001001                          BNE     2$              ;/BR IF YES NEXT TESt.

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

013020  104012                          ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                ;/COUNT RATE: 1MHZ.

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

013022                          2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 101      *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"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
013022  000004                          TST101: SCOPE
013024  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/MAKE SURE CLOCK IS CLEAR.
                                                                        ;/CLEAR THE BUFFER.
                                                                        ;/SELECT: MODE 0, RATE 100KHZ ; GO.
013032  005037  001124                       CLR     $GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
013056  012737  000005  001124                MOV     #1!4,$GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
013074  005000                               CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
013076  005200                          1$:  INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
013100  001376                               BNE     1$
                                                                        ;/DID COUNTER INCREMENT AT ALL?

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
013112  005737  001126                       TST     $BDDAT
013116  001011                               BNE     2$              ;/IF YES - BR NEXT TEST.

                                                                        ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.

                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
013130  032737  000040  001126                BIT     #BIT05,$BDDAT
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
013136  001001                               BNE     2$              ;/BR IF YES NEXT TEST.

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

013140  104012                               ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: 100KHZ.

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

013142                                  2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 102      *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"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
013142  000004                          TST102: SCOPE
013144  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/MAKE SURE CLOCK IS CLEAR.
                                                                        ;/CLEAR THE BUFFER.
                                                                        ;/SELECT: MODE 0, RATE 10KHZ ; GO.
013152  005037  001124                      CLR     $GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
013176  012737  000007  001124          1$:     MOV     #1!6,$GDDAT
                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
013214  005000                          1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
013216  005200                          1$:     INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
013220  001376                          1$:     BNE     1$
                                                                        ;/DID COUNTER INCREMENT AT ALL?
                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
013232  005737  001126                      TST     $BDDAT
013236  001011                              BNE     2$                  ;/IF YES - BR NEXT TEST.
                                                                        ;/COUNTER MAY HAVE HAD TIME TO oVERFLOW.
                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
013250  032737  000040  001126                BIT     #BIT05,$BDDAT
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
013256  001001                              BNE     2$                  ;/BR IF YES NEXT TEST.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

013260  104012                              ERROR   12                  ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: 10KHZ.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

013262                          2$:
```

```
                                                ;/*
                                        ;;*****************************************************************
                                        ;*TEST 103      *TEST THE ABILITY OF CLOCK A TO COUNT AT 1KHZ RATE PART 1
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO TEST CLOCK A'S ABILITY TO COUNT
                                        ;*IN RATE: 1KHZ          PART 1
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
 013262  000004                         TST103: SCOPE
 013264  012737  000020  001166                 MOV     #20,$TIMES              ;;DO 20 ITERATIONS

                                                                                ;/MAKE SURE CLOCK IS CLEAR.
                                                                                ;/CLEAR THE BUFFER.
                                                                                ;/SELECT: MODE 0, RATE 1KHZ ; GO.
 013272  005037  001124                         CLR     $GDDAT

                                        ;*      MOV     $GDDAT,@ASR             ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     $GDDAT,@ABR             ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
 013316  012737  000011  001124                 MOV     #1!10,$GDDAT

                                        ;*      MOV     $GDDAT,@ASR             ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
 013334  005000                                 CLR     R0                      ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
 013336  005200                         1$:     INC     R0                      ;/WILL AMOUNT TO 369MS ON A PDP-11/20
 013340  001376                                 BNE     1$
                                                                                ;/DID COUNTER INCREMENT AT ALL?

                                        ;*      MOV     @ACR,$BDDAT             ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
 013352  005737  001126                         TST     $BDDAT
 013356  001011                                 BNE     2$                      ;/IF YES - BR NEXT TEST.

                                                                                ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.

                                        ;*      MOV     @ASR,$BDDAT             ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
 013370  032737  000040  001126                 BIT     #BIT05,$BDDAT
                                                                                ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                                ;/F/F HAD SET.
 013376  001001                                 BNE     2$                      ;/BR IF YES NEXT TEST.

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

 013400  104012                                 ERROR   12                      ;/ERROR CLOCK A COUNTER FAILED TO
                                                                                ;/COUNT RATE: 1KHZ.

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

 013402                                 2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 104       *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"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
013402  000004                        TST104: SCOPE
013404  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/MAKE SURE CLOCK IS CLEAR.
                                                                        ;/CLEAR THE BUFFER.
                                                                        ;/SELECT: MODE 0, RATE 100HZ ; GO.
013412  005037  001124                      CLR     $GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
013436  012737  000013  001124              MOV     #1!12,$GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
013454  005000                              CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY: THIS DELAY
013456  005200                        1$:   INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
013460  001376                              BNE     1$
                                                                        ;/DID COUNTER INCREMENT AT ALL?

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
013472  005737  001126                      TST     $BDDAT
013476  001011                              BNE     2$              ;/IF YES - BR NEXT TEST.

                                                                        ;/COUNTER MAY HAVE HAD TIME TO oVERFLOW.

                                        ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
013510  032737  000040  001126              BIT     #BIT05,$BDDAT
                                                                        ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                        ;/F/F HAD SET.
013516  001001                              BNE     2$              ;/BR IF YES NEXT TESt.

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

013520  104012                              ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                        ;/COUNT RATE: 100HZ.

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

013522                                  2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 105       *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"
                                ;*
                                ;*
                                ;;**********************************************************************
013522  000004                  TST105: SCOPE
013524  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                ;/MAKE SURE CLOCK IS CLEAR.
                                                                ;/CLEAR THE BUFFER.
                                                                ;/SELECT: MODE 0, RATE LINE-FREQ ; GO.
013532  005037  001124          CLR     $GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
013556  012737  000017  001124          MOV     #1!16,$GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
013574  005000                  1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY; THIS DELAY
013576  005200                          INC     R0              ;/WILL AMOUNT TO 369MS ON A PDP-11/20
013600  001376                          BNE     1$
                                                                ;/DID COUNTER INCREMENT AT ALL?
                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
013612  005737  001126                  TST     $BDDAT
013616  001011                          BNE     2$              ;/IF YES - BR NEXT TEST.
                                                                ;/COUNTER MAY HAVE HAD TIME TO OVERFLOW.
                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
013630  032737  000040  001126          BIT     #BIT05,$BDDAT
                                                                ;/AT HIGH RATE - SO WE'LL SEE IF "OVERFLOW"
                                                                ;/F/F HAD SET.
013636  001001                          BNE     2$              ;/BR IF YES NEXT TEST.

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

013640  104012                          ERROR   12              ;/ERROR CLOCK A COUNTER FAILED TO
                                                                ;/COUNT RATE: LINE-FREQ.

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

013642                          2$:
```

```
                                ;;**********************************************************************
                                ;*TEST 106      *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"
                                ;*
                                ;;**********************************************************************
013642  000004                  TST106: SCOPE
013644  012737  000050  001166          MOV     #50,$TIMES      ;;DO 50 ITERATIONS

                                                                ;CLEAR CLOCK A.
                                                                ;ZERO ITS BUFFER.
                                                                ;TELL THE CLOCK TO GO!
013652  005037  001124          CLR     $GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
013676  005237  001124          INC     $GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                                                ;NO RATE SELECTED.
013712  005000          CLR     R0
013714  005200  1$:     INC     R0
013716  001376          BNE     1$
                                                                ;ANY COUNT OCCUR?
                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
013730  005737  001126          TST     $BDDAT
013734  001010          BNE     2$              ;IF COUNTER HAS SOMETHING IN IT REPORT ERROR

                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
013746  032737  000040  001126          BIT     #BIT05,$BDDAT   ;IF THE OVERFLOW BIT SET THEN IT MUST
                                                                ;HAVE COUNTED.
013754  001401          BEQ     3$

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

013756  104012  2$:     ERROR   12              ;AH HA! ERROR CLOCK A COUNTED WHEN
                                                ;ENABLED-BUT-NO-RATE SELECTED
                                                ;BETTER FIND OUT HOW CAUSED SIGNAL:
                                                ;"CLOCK A" L.

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

```
                                3$:
                                ;;*****************************************************************
                                ;*TEST 107      *TEST THAT CLOCK A'S COUNT REG ISN'T LOADED WHEN CLOCK A IS ENABLED
                                ;*
                                ;*IN THIS TEST WE'LL FIND OUT IF F/F "ENB CNTR A" WHEN SET,
                                ;*INHIBITS LOADING OF CLOCK A'S COUNT REGISTER
                                ;*THE LOADING OF THE COUNT REGISTER IS A FUNCTION OF:
                                ;*"ENB CNTR (0)" H + "BUFFER LOAD" H
                                ;*
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "RD STAT B"
                                ;*
                                ;;*****************************************************************
013760  000004                  TST107: SCOPE
                                                                ;GENERATE A SYNC PULSE
                                                                ;CLEAR CLOCK A'S STATUS REG.
                                                                ;CLEAR CLOCK A'S BUFFER REG. NOTE
                                                                ;THIS WILL ALSO CAUSE ZEROS TO BE
                                                                ;LOADED INTO THE COUNT REG.
                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
013772  005737  001126                  TST     $BDDAT
013776  005037  001124                  CLR     $GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
014022  005237  001124                  INC     $GDDAT          ;SET THE ENABLE F/F. THIS
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                                                ;FROM BEING LOADED WHEN THE
                                                                ;BUFFER IS LOADED.
014036  012737  177777  001124          MOV     #177777,$GDDAT  ;NOW LOAD THE BUFFER REG.
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                                                ;TO THE COUNT REGISTER.
                                                                ;DID ANY BITS GET TRANSFERRED TO
                                                                ;THE COUNT REGISTER?
                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
014064  005737  001126                  TST     $BDDAT
014070  001401                          BEQ     1$              ;BR IF NO - NEXT TEST.
                        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

014072  104012                          ERROR   12              ;ERROR CLOCK A BUFFER TO COUNT REG TRANSFER
                                                                ;OCCURRED EVEN THOUGH THE "ENB CNTR A" F/F
                                                                ;WAS SET.
                        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
4572  014074                          1$:
4573
4574
4575                                  ;;*****************************************************************
4576                                  ;*TEST 110      *TEST THAT CLOCK A IN MODE 1 DOES NOT CLEAR ENABLE ON OVERFLOW
4577                                  ;*
4578                                  ;*NOW WE'RE GOING TO SEE IF "A OVERFLOW" H WHEN GENERATED
4579                                  ;*BY CAUSING AN OVERFLOW DOESN'T CLEAR THE "ENB CNTR A" F/F
4580                                  ;*WHEN IN MODE 1. IF F/F GETS CLEARED SUSPECT SIGNAL "MODE 0" H.
4581                                  ;*
4582                                  ;*
4583                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
4584                                  ;*
4585                                  ;;*****************************************************************
4586  014074  000004                  TST110: SCOPE
4587
4588                                                                  ;MAKE SURE CLOCK A IS CLEAR.
4589                                                                  ;SET BUFFER + COUNT REG. TO -1 FROM OVERFLOW
4590                                                                  ;SET: MODE 1; RATE STP1; GO.
4591                                                                  ;CAUSE A MAINTENANCE STP4. CLOCK 1
4592                                                                  ;SHOULD OVERFLOW - BUT THIS OVERFLOW SHOULD
4593                                                                  ;NOT CLEAR "ENB CNTR A" F/F.
4594                                                                  ;DID BIT00, "ENB CNTR A" F/F GET CLEARED?
4595  014076  005037  001124                  CLR     $GDDAT
4596
4597                                  ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEViCE REG ASR
4598  014112  012737  177777  001124          MOV     #177777,$GDDAT
4599
4600                                  ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
4601  014130  012737  000415  001124          MOV     #415,$GDDAT
4602
4603                                  ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4604
4605                                  ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
4606  014156  052737  010000  001356          BIS     #BIT12,$TMDAT
4607
4608                                  ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO dEVICE REG ASR
4609
4610                                  ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
4611  014204  032737  000001  001126          BIT     #BIT00,$BDDAT
4612  014212  001001                          BNE     1$              ;BR IF NO TO NEXT TESt.
4613
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

4617  014214  104012                          ERROR   12              ;ERROR MODE 1 OPERATION "ENB CNTR A" F/F
4618                                                                  ;WAS CLEARED ON OVERFLOW.
4619
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

4623  014216                        1$:
```

```
                        ;*****************************************************************
                        ;*TEST 111      *TEST THAT A CLOCK A "BUFFER TO COUNT REG" DOESN'T TAKE PLACE ON A MODE
                        ;*
                        ;*THIS WILL BE THE FIRST TIME WE'VE DONE A MODE 2 OPERATION
                        ;*ON CLOCK A. THE FUNCTION OF THIS TESt WILL BE TO MAKE
                        ;*SURE A BUFFER TO COUNT REGISTER DOESN'T TAKE PLACE ON OVERFLOW
                        ;*THE COMBO THAT GAVE US BUFFER TO COUNT REG ON OVERFLOW BEFORE
                        ;*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.
                        ;*
                        ;*****************************************************************
014216  000004          TST111: SCOPE

                                                        ;MAKE SURE CLOCK A IS CLEAR.
                                                        ;SET BUFFER + COUNT REG TO -1 FROM OVERFLOW.
                                                        ;SET: MODE 2; RATE STP1; GO.
                                                        ;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.
                                                        ;IS THE COUNT REG. CLEAR?
014220  005037  001124          CLR     $GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
014234  012737  177777  001124        MOV     #177777,$GDDAT
                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
014252  012737  001015  001124        MOV     #1015,$GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
014300  052737  010000  001124        BIS     #BIT12,$GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
014326  005737  001126          TST     $BDDAT
014332  001401                  BEQ     1$              ;BR IF YES TO NEXT TEST.

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

014334  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.

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

014336                          1$:

                                ;;**********************************************************************
                                ;*TEST 112       *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"
                                ;*
                                ;*
                                ;;**********************************************************************
014336  000004                TST112: SCOPE

                                                        ;GENERATE A SYNC PULSE

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
014350  005737  001126                  TST     $BDDAT
                                                        ;MAKE SURE CLOCK A'S STAT REG IS CLEAR.
014354  005037  001124                  CLR     $GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                                        ;SET MODE 2.
                                                        ;GENERATE MAINTENANCE ST2.
                                                        ;THE COMBO OF MODE 2 + ST2 SHOULD
                                                        ;GET "CNTR TO BUFF" H WHICH SHOULD
                                                        ;SET "MODE FLG" F/F.
014370  012737  001000  001124          MOV     #1000,$GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
014416  052737  002000  001124          BIS     #BIT10,$GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                                        ;DID IT SET?

                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
014444  032737  000200  001126          BIT     #BIT07,$BDDAT
014452  001001                  BNE     1$              ;IF YES-BR TO NEXT TEST.
```

```
4732
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4736  014454  104012                          ERROR   12              ;ERROR - MODE 2 + ST2 DID NOT SET MODE FL.
4737
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

4741  014456                          1$:


                                        ;*********************************************************************
                                        ;*TEST 113      *TEST THAT PATTERN 052525 CAN BE XFERRED BETWEEN A'S COUNT-BUFFER REGS
                                        ;*
                                        ;*NOW WE'LL SHOT THE WORKS - WE KNOW FROM THE PREVIOUS TEST WE
                                        ;*CAN GENERATE "CNTR TO BUFF" H FROM MODE 2 + MAINTENANCE ST2, NOW
                                        ;*WE WELL TRY AND GENERATE A TRANSFER BETWEEN THE COUNTER AND BUFFER
                                        ;*USING A CB PAT PATTERN.
                                        ;*IF NO DATA PATTERN GETS TRANSFERRED, SUSPECT SIG "LD BUFFER"
                                        ;*TO BE STUCK LOW AT THE MUX INPUT FOR THE BUFFER OR
                                        ;*"CNTR TO BUFF" H NOT GETTING THROUGH TO THE LOAD INPUTS OF THE
                                        ;*BUFFER REGISTER.
                                        ;*IF JUST ONE OR A FEW BITS GETS MESSED UP ON THE XFERR-
                                        ;*SUSPECT THE RESPECTIVE MUX OR ETCH BETWEEN THE COUNT REG
                                        ;*AND MUX. GOOD LUCK.
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "RD STAT B"
                                        ;*
                                        ;*
                                        ;*********************************************************************
4763  014456  000004                    TST113: SCOPE

                                                                ;GENERATE A SYNC PULSE

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
4768  014470  005737  001126                  TST     $BDDAT

                                                                ;MAKE SURE CLOCK A IS CLEAR.
                                                                ;PUT PATTERN 052525 INTO BUFFER REG.
                                                                ;IT SHOULD GET XFERRED TO COUNT REG.
                                                                ;SELECT: MODE 2, ENABLE.
                                                                ;NOW GENERATE A MAINTENANCE ST2.
4774  014474  005037  001124                  CLR     $GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4777  014510  012737  052525  001124        MOV     #052525,$GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
4780  014526  012737  001001  001124        MOV     #1001,$GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4783  014544  005037  001124                  CLR     $GDDAT

                                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
```

```
                                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
014570  052737  002000  001124          ;       BIS     #BIT10,$GDDAT

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
014606  012737  052525  001124          ;       MOV     #052525,$GDDAT  ;RECORD $GDDAT (PATTERN) IN CASE WE
                                                                        ;NEED TO TYPE OUT AN ERROR.
                                                                        ;NOW READ BACK THE BUFFER REG.

                                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
014624  023737  001126  001124          ;       CMP     $BDDAT,$GDDAT   ;WAS THE TRANSFER SUCCESSFUL?
014632  001401                          ;       BEQ     1$              ;IF YES THEN BR TO NEXT TEST.

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

014634  104012                          ;       ERROR   12              ;ERROR FAILED TO XFERR 052525 PATTERN
                                                                        ;CORRECTLY FROM COUNT TO BUFFER REG.
                                                                        ;SEE INIT. COMMENT AS TO WHY
                                                                        ;IT MIGHT HAVE GONE SOUR.

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

014636                          1$:
000011                                  P=P+1

                                ;;*****************************************************************
                                ;*TEST 114      *TEST THAT PATTERN 125252 CAN BE XFERRED BETWEEN A'S COUNT-BUFFER REGS
                                ;*
                                ;*NOW WE'LL SHOT THE WORKS - WE KNOW FROM THE PREVIOUS TEST WE
                                ;*CAN GENERATE "CNTR TO BUFF" H FROM MODE 2 + MAINTENANCE ST2, NOW
                                ;*WE WELL TRY AND GENERATE A TRANSFER BETWEEN THE COUNTER AND BUFFER
                                ;*USING A CB PAT PATTERN.
                                ;*IF NO DATA PATTERN GETS TRANSFERRED, SUSPECT SIG "LD BUFFER"
                                ;*TO BE STUCK LOW AT THE MUX INPUT FOR THE BUFFER OR
                                ;*"CNTR TO BUFF" H NOT GETTING THROUGH TO THE LOAD INPUTS OF THE
                                ;*BUFFER REGISTER.
                                ;*IF JUST ONE OR A FEW BITS GETS MESSED UP ON THE XFERR-
                                ;*SUSPECT THE RESPECTIVE MUX OR ETCH BETWEEN THE COUNT REG
                                ;*AND MUX. GOOD LUCK.
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "RD STAT B"
                                ;*
                                ;*
                                ;*****************************************************************
014636  000004                  TST114: SCOPE
                                                                        ;GENERATE A SYNC PULSE

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
014650  005737  001126                  TST     $BDDAT
                                                                        ;MAKE SURE CLOCK A IS CLEAR.
                                                                        ;PUT PATTERN 125252 INTO BUFFER REG.
```

```
                                                        ;IT SHOULD GET XFERRED TO COUNT REG.
                                                        ;SELECT: MODE 2, ENABLE.
                                                        ;NOW GENERATE A MAINTENANCE ST2.
014654  005037  001124                  CLR     $GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
014670  012737  125252  001124          MOV     #125252,$GDDAT
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
014706  012737  001001  001124          MOV     #1001,$GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
014724  005037  001124                  CLR     $GDDAT
                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
                                ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
014750  052737  002000  001124          BIS     #BIT10,$GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
014766  012737  125252  001124          MOV     #125252,$GDDAT  ;RECORD $GDDAT (PATTERN) IN CASE WE
                                                        ;NEED TO TYPE OUT AN ERROR.
                                                        ;NOW READ BACK THE BUFFER REG.
                                ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
015004  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;WAS THE TRANSFER SUCCESSFUL?
015012  001401                          BEQ     1$              ;IF YES THEN BR TO NEXT TEST.

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

015014  104012                  ERROR   12                      ;ERROR FAILED TO XFERR 125252 PATTERN
                                                        ;CORRECTLY FROM COUNT TO BUFFER REG.
                                                        ;SEE INIT. COMMENT AS TO WHY
                                                        ;IT MIGHT HAVE GONE SOUR.

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

015016                          1$:
        000012                          P=P+1


                                ;;**********************************************************************
                                ;*TEST 115      *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"
                                ;*
                                ;*
                                ;;**********************************************************************
```

```
4894  015016  000004                    TST115: SCOPE
4895
4896                                                          ;MAKE SURE CLOCK A IS CLEAR.
4897                                                          ;LOAD ALL ONES INTO BUFFER COUNT REGS.
4898                                                          ;SET MODE 1.
4899                                                          ;GENERATE A MAINTENANCE STP2.
4900                                                          ;SEE IF IT CLEARED COUNT REG.
4901  015020  005037  001124                    CLR     $GDDAT
4902
4903                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4904  015034  012737  177777  001124            MOV     #177777,$GDDAT
4905
4906                                    ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
4907  015052  012737  000200  001124            MOV     #200,$GDDAT
4908
4909                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4910
4911                                    ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
4912  015100  052737  002000  001124            BIS     #BIT10,$GDDAT
4913
4914                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
4915
4916                                    ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
4917  015126  005737  001126                    TST     $BDDAT
4918  015132  001001                            BNE     1$              ;BR IF NO - NEXT TEST.
4919
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

4923  015134  104012                    ERROR   12                      ;ERROR CLOCK A MODE 1 + STP2 CLEARED COUNT REG.
4924                                                          ;TO SCOPE FIND OUT WHAT IS GENERATING
4925                                                          ;SIGNAL ON CLR INPUT OF COUNT REG.
4926
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

4930  015136                            1$:
4931
4932                                    ;;*****************************************************************
4933                                    ;*TEST 116      *TEST THAT A'S COUNT REGISTER ISN'T CLEARED IN MODE 2 WHEN STP2 IS GENER
4934                                    ;*
4935                                    ;*THIS TEST IS DESIGNED TO MAKE SURE THAT WE DON'T CLEAR THE COUNT
4936                                    ;*REGISTER IN MODE 2 WHEN AN STP2 IS GENERATED.
4937                                    ;*
4938                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
4939                                    ;*
4940                                    ;*
4941                                    ;*****************************************************************
4942  015136  000004                    TST116: SCOPE
4943
4944                                                          ;MAKE SURE CLOCK A IS CLEAR.
4945                                                          ;LOAD ALL ONES INTO BUFFER COUNT REGS.
4946                                                          ;SET MODE 2.
4947                                                          ;GENERATE A MAINTENANCE STP2.
```

```
                                                        ;SEE IF IT CLEARED COUNT REG.
015140  005037  001124          CLR     $GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
015154  012737  177777  001124  MOV     #177777,$GDDAT
                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
015172  012737  001000  001124  MOV     #1000,$GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
015220  052737  002000  001124  BIS     #BIT10,$GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
015246  005737  001126          TST     $BDDAT
015252  001001                  BNE     1$              ;BR IF NO - NEXT TEST.

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

015254  104012          ERROR   12                      ;ERROR CLOCK A MODE 2 + STP2 CLEARED COUNT REG.
                                                        ;TO SCOPE FIND OUT WHAT IS GENERATING
                                                        ;SIGNAL ON CLR INPUT OF COUNT REG.

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

015256                  1$:


                        ;;*****************************************************************
                        ;*TEST 117      *TEST THAT MODE 3 + "STP2" CLEARS A'S COUNT REGISTER.
                        ;*
                        ;*IN THIS TEST WE'LL DO A MODE 3 FOR THE FIRST TIME.
                        ;*MODE 3 + "STP2" SHOULD CLEAR THE COUNT REGISTER.
                        ;*WE KNOW FROM A PREVIOUS TEST THAT "INIT" L WAS ABLE TO
                        ;*CLEAR THE REG. AND ALSO THAT WE CAN GENERATE AN "STP2" H.
                        ;*"MODE A0 (1)" H + "MODE A1 (1)" H + "STP2" H = CLEARING SIGNAL.
                        ;*
                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                        ;*
                        ;*
                        ;;*****************************************************************
015256  000004          TST117: SCOPE

                                                        ;CLEAR CLOCK A + SELECT MODE 3.
                                                        ;LOAD ALL ONE'S TO BUFFER + COUNTER REGS.
                                                        ;GENERATE A MAINTENANCE "STP2".
                                                        ;THIS SHOULD CLEAR THE COUNT REG.

                                                        ;IS COUNT REG. CLEAR?
```

```
5002  015260  012737  001400  001124          MOV     #1400,$GDDAT
5003
5004                                  ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
5005  015276  012737  177777  001124          MOV     #177777,$GDDAT
5006
5007                                  ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
5008
5009                                  ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
5010  015324  052737  002000  001124          BIS     #BIT10,$GDDAT
5011
5012                                  ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
5013
5014                                  ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
5015  015352  005737  001126                  TST     $BDDAT
5016  015356  001401                          BEQ     1$              ;BR IF YES - NEXT TEST.
5017
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5021  015360  104012                          ERROR   12              ;ERROR-CLOCK A-COUNT REGISTER FAILED TO
5022                                                                  ;CLEAR IN MODE 3 ON "STP2" PULSE.
5023
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5027  015362                          1$:

5028
5029
5030                                  ;;*****************************************************************
5031                                  ;*TEST 120       *TEST THE AUTODECREMENT FEATURE OF CLOCK A'S BUFFER
5032                                  ;*
5033                                  ;*THIS IS GOING TO BE A BIGGIE-WITH TWO PARTS.
5034                                  ;*PART1:         WE'RE LOADING THE BUFFER WITH ALL ONES; MODE 0; RATE STP1,
5035                                  ;*               WITH "AUTO INC (1)" SET (FOR FIRST TIME!). NOW WE'LL GENERATE
5036                                  ;*               THE THING TO WATCH FOR IS THE EXPLOSIVE COMBO OF
5037                                  ;*               "MODE A0 (0)" H + "MODE A1 (0)" H (MODE 0) + "AUTO INC (1)" H +
5038                                  ;*               "A OVERFLOW" ALL FEEDING THE DOWn COUNT SIDE OF THE 74193 CHIP.
5039                                  ;*               THE RESULTS SHOULD BE 177776.
5040                                  ;*PART2:         IF PART 1 WAS SUCCESSFUL WE'LL LOOK TO SEE IF
5041                                  ;*               THE COUNTER GOT LOADED WITH THE NEW VALUE 177776.
5042                                  ;*               THE HANG-UP HERE IS THAT "A OVERFLOW" FEEDS A ONE SHOT
5043                                  ;*               THAT GENERATES "A RELOAD" H. WE KNOW THAT WE CAN
5044                                  ;*               GET "A RELOAD" H BUT THE BIG TEST HERE IS SEEING IF THAT
5045                                  ;*               ONE SHOT SPITS OUT "A RELOAD" H TOO SOON TO CATCH THE BUFFER
5046                                  ;*               WITH ITS PANTS DOWN AS ITS DECREMENTING ITSELF.
5047                                  ;*
5048                                  ;*
5049                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
5050                                  ;*
5051                                  ;*****************************************************************
5052  015362  000004                  TST120: SCOPE
5053
5054                                                          ;MAKE SURE CLOCK B IS CLEAR.
5055                                                          ;MAKE SURE CLOCK A IS CLEAR.
```

```
                                                        ;LOAD BUFFER + COUNT REGS.
                                                        ;SET AUTO INCREMENT MODE
                                                        ;ENABLE COUNTER, MODE 0, RATE STP1
                                                        ;ZAP! A MAINTENANCE STP1 HAS BEEN MADE.
                                                        ;STOP HERE AND LOOK WHAT JUST HAPPENED.
                                                        ;1. STP1 CLOCKED THE COUNTER (SET 177777).
                                                        ;2. COUNTER OVERFLOWED - "A OVERFLOW" L
                                                        ;3. MODE 0 + "A OVERFLOW" L + "A AUTO INC (1)" H
                                                        ;   DOWN COUNTED THE BUFFER.
                                                        ;4. "A OVERFLOW" L GOES THROUGH ONE SHOT DELAY
                                                        ;   TO GIVE "A RELOAD" H
                                                        ;5. "A RELOAD" H CAUSES A BUFFER TO COUNTER
                                                        ;   RELOAD.
015364  005037  001124          CLR     $GDDAT
                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
015410  012737  177777  001124  MOV     #177777,$GDDAT
                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
015426  012737  000020  001124  MOV     #20,$GDDAT
                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
015444  012737  000015  001124  MOV     #15,$GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
015472  052737  010000  001124  BIS     #BIT12,$GDDAT
                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                        ;PART 1 DID BUFFER GET DECREMENTED?

015510  012737  177776  001124  MOV     #177776,$GDDAT  ;SET FOR ERROR TYPEOUT IF ANY.
                                                        ;READ THE RESULTS OF THE BUFFER.
                        ;*      MOV     @ABR,$BDDAT     ;/READ DEVICE REG ABR,PUT DATA IN $BDDAT.
015526  023727  001126  177776  CMP     $BDDAT,#177776  ;DID BUFFER DECREMENT TO 177776?
015534  001402                  BEQ     1$              ;BR IF YES TO PART 2

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

015536  104012                  ERROR   12              ;ERROR-CLOCK A. AFTER AN OVERFLOW
                                                        ;WITH AUTO INC ENABLED MODE 0, THE
                                                        ;BUFFER REGISTER FAILED TO DOWN COUNT
                                                        ;SEE ABOVE COMMENTS FOR SEQUENCE
                                                        ;OF EVENTS.
015540  000411                  BR      2$              ;IF ERROR ONLY LOOP ON PART 1.

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

```
5111  015542                          1$:     ;PART 2 DID NEW COUNT GET TRANSFERRED TO COUNT REGISTER?
5112
5113                                                                  ;READ THE COUNT REGISTER
5114
5115                                  ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
5116  015552  023727  001126  177776         CMP     $BDDAT,#177776  ;DID NEW VALUE OF THE BUFFER
5117                                                                  ;REGISTER GET PROPERLY LOADED INTO
5118                                                                  ;THE COUNT REGISTER?
5119  015560  001401                        BEQ     2$              ;BR IF YES - NEXT TEST.
5120
          ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5124  015562  104012                        ERROR   12              ;ERROR CLOCK A. NEW CONTENTS OF
5125                                                                  ;BUFFER REGISTER FAILED TO BE PROPERLY
5126                                                                  ;LOADED INTO COUNT REGISTER AFTER
5127                                                                  ;AUTO DECREMENT.
5128                                                                  ;SEE ABOVE COMMENTS FOR SEQUENCE OF EVENTS.
5129
          ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5133  015564                          2$:                             ;CLEAR AUTO INC OPTION.
5134  015564  005037  001124                 CLR     $GDDAT
5135
5136                                  ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
5137
5138                                  ;;**************************************************************
5139                                  ;*TEST 121       *TEST THAT CLOCK A'S 1 MHZ CLK CAN BE DISABLED
5140                                  ;*
5141                                  ;*FOR THIS TEST, WE'LL TRY DISABLING THE 1 MHZ CLOCK. TO DO THIS,
5142                                  ;*WE'LL SET THE "DISABLE OSC 1 MHZ" BIT 11 IN CLOCK B SR. THEN
5143                                  ;*COUNTED OR OVERFLOWED, IF SO ERROR.
5144                                  ;*THE UNKNOWN THING HERE IS BIT 11 SETTING THE DISABLE F/F THAT
5145                                  ;*GATES WITH THE 1 MHZ FREG TO pRODUCE "A 1 MHZ CLK" L
5146                                  ;*
5147                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
5148                                  ;*
5149                                  ;;**************************************************************
5150  015600  000004                  TST121: SCOPE
5151  015602  012737  000100  001166         MOV     #100,$TIMES     ;;DO 100 ITERATIONS
5152
5153                                                                  ;CLEAR CLOCK A.
5154                                                                  ;SET THE "DISABLE OSC 1 MHZ" F/F.
5155                                                                  ;CLEAR THE BUFFER + COUNT REGS.
5156                                                                  ;START CLOCK: RATE 1 MHZ, MODE 0, GO.
5157  015610  005037  001124                 CLR     $GDDAT
5158
5159                                  ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
5160  015624  012737  004000  001124         MOV     #BIT11,$GDDAT
5161
5162                                  ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
5163  015642  005037  001124                 CLR     $GDDAT
```

```
5164
5165                                    ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
5166  015656  012737  000003  001124            MOV     #3,$GDDAT
5167
5168                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
5169  015674  005000                            CLR     R0              ;SHORT DELAY. WE ALLOW THIS DELAY TO
5170  015676  105200                    1$:     INCB    R0              ;OCCUR. IF THE 1 MHZ CLOCK IS DISABLED
5171  015700  100376                            BPL     1$              ;NO CLOCKING OF CLOCK A WILL OCCUR.
5172                                                                    ;SEE SOMETHING IN THE COUNT REGISTERS
5173
5174                                    ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
5175  015712  005737  001126                    TST     $BDDAT          ;DOES COUNT REG HAVE ANYTHING IN IT?
5176  015716  001007                            BNE     2$              ;YES - REPORT ERROR!
5177
5178                                    ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
5179  015730  105737  001126                    TSTB    $BDDAT
5180  015734  100001                            BPL     3$              ;NO - BR NEXT TEST - NO ERROR.
5181
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

5185  015736  104012                    2$:     ERROR   12              ;ERROR - UNABLE TO DISABLE 1 MHZ CLK
5186                                                                    ;USING "DISABLE OSC 1 MHZ" F/F
5187
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

5191                                                                    ;CLEAR CLOCK A.
5192  015740  005037  001124            3$:     CLR     $GDDAT
5193
5194                                    ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
```

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


                                ;;**************************************************************
                                ;*TEST 122       *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"
                                ;*
                                ;*
                                ;;**************************************************************
015754  000004                  TST122: SCOPE

                                                                ;CLEAR CLOCK B AND DISABLE 1MHZ OSC.
015756  012737  004000  001124          MOV     #BIT11,$GDDAT
                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                                                ;CLEAR B'S BUFFER + COUNT REGS.
                                                                ;SELECT "FEED B TO A", RATE 1 MHZ, GO.
015774  005037  001124          CLR     $GDDAT
                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                                                ;CLOCK A'S 1 MHZ CLOCK.
                                                                ;GENERATE A MAINTENANCE 1 MHZ PULSE..
                                                                ;DID COUNTER COUNT?
                                ;*      MOV     @BSR,$GDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $GDDAT.
016020  052737  000043  001124          BIS     #BIT5!BIT1!BIT0,$GDDAT
                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
016046  052737  004000  001124          BIS     #BIT11,$GDDAT
                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
016074  005737  001126          TST     $BDDAT
016100  001001                  BNE     1$              ;BR IF YES TO NEXT TESt.

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

```
                                                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.

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

016104                                  1$:


                                        ;;**************************************************************
                                        ;*TEST 123      *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"
                                        ;*
                                        ;**************************************************************
016104  000004                  TST123: SCOPE
016106  012737  000001  001166          MOV     #1,$TIMES       ;;DO 1 ITERATION

016114  005037  001124          1$:     CLR     $GDDAT  ;START THE COUNTER FROM ZERO.
                                                                ;NOTE: A VALUE OTHER THAN ZERO MAY BE
                                                                ;PATCHED IN HERE IN ORDER THAT A COUNT
                                                                ;MAY BE STARTED HIGHER.
016120  005037  001356          1$:     CLR     $TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
                                                                ;CLEAR CLOCK B, DISABLE 1 MHZ CLOCK A.
016134  012737  004000  001356          MOV     #BIT11,$TMDAT

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
                                                                ;LOAD VALUE INTO COUNT + BUFFER REGS.
                                                                ;"1$" IS THE LOOP BACK POINT ON
                                                                ;"LOOP ON TEST" (SW14=1) FEATURE. NORMAL
                                                                ;LOOP BACK POINT WILL BE "2$".
```

```
                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
                                                                        ;SELECT: "DISABLE OSC 1 MHZ"; "FEED B TO A";
                                                                        ;RATE 1 MHZ.
                                                                        ;GO. ENABL MUST BE SET AFTER "FEED B TO A"
016162  012737  004042  001356          MOV     #4042,$TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
016200  005237  001356                  INC     $TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

016214                                  2$:                             ;GENERATE A CLOCK PULSE.
                                                                        ;SHOULD CAUSE THE CLOCK TO
                                                                        ;INCREMENT ONCE.

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
016224  052737  004000  001356          BIS     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
016242  005237  001124                  INC     $GDDAT          ;$GDDAT IS USED TO KEEP TRACK OF THE
                                                                ;VALUE THE CLOCK SHOULD COUNT TO.

                                        ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.

016256  123737  001126  001124          CMPB    $BDDAT,$GDDAT   ;DID THE COUNT OCCUR CORRECTLY?
016264  001402                          BEQ     3$              ;IF YES - BR "3$".

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

016266  104015                          ERROR   15              ;ERROR CLOCK B FAILED TO UPCOUNT
                                                                ;CORRECTLY - SEE COMMENTS AT SUB-TEST
                                                                ;HEADING FOR MORE DETAILS.

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

016270  000403                          BR      4$

016272  105737  001124          3$:     TSTB    $GDDAT          ;DID WE ACHIEVE OVERFLOW YET?
016276  001410                          BEQ     5$              ;

016300  032777  040000  162632  4$:     BIT     #BIT14,@SWR     ;LOOP ON CURRENT COUNT?
016306  001742                          BEQ     2$              ;BR IF NO TO NEXT COUNT UPDATE.
016310  162737  000001  001124          SUB     #1,$GDDAT       ;IF YES DECREMENT EXPECTED AND RELOAD.
016316  000700                          BR      1$              ;GO TO rELOAD POINT

016320                                  5$:                     ;END SUBTEST.


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

```
                                    ;*TEST 124      *TEST THAT CLOCK B CAN GENERATE AN OVERFLOW
                                    ;*
                                    ;*NOW WE'LL TRY AND GENERATE "B OVERFLOW" L WHICH SETS BIT 07.
                                    ;*WE'LL DO IT BY PRESETTING THE BUFFER TO 377 AND GENERATING A "CLOCK B" L
                                    ;*WE ALREADY KNOW WE CAN ADVANCE THE COUNTER, WHAT WE
                                    ;*WANT TO SEE IS "B OVERFLOW" L COME OVER AND DIRECT SET
                                    ;*"OVERFLOW FL B" F/F (BIT 07 IN CSR).
                                    ;*
                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
                                    ;*
                                    ;*
                                    ;;*****************************************************************
016320  000004                      TST124: SCOPE

                                                          ;CLEAR CLOCK A.
                                                          ;CLEAR CLOCK B.
                                                          ;SET COUNT AND BUFFER REGS.
                                                          ;SELECT "DISABLE OSC 1 MHZ"; "FEED B TO A";
                                                          ;RATE 1 MHZ.
                                                          ;GO. ENABL MUST BE SET AFTER "FEED B TO A"
                                                          ;GENERATE A CLOCK PULSE.

                                                          ;DID OVERFLOW FLAG SET?
016322  005037  001124                    CLR     $GDDAT
                                ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*        MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
016346  012737  000377  001124            MOV     #377,$GDDAT
                                ;*        MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
016364  012737  004042  001124            MOV     #4042,$GDDAT
                                ;*        MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
016402  005237  001124                    INC     $GDDAT
                                ;*        MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                ;*        MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
016426  052737  004000  001124            BIS     #BIT11,$GDDAT
                                ;*        MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*        MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
016454  105737  001126                    TSTB    $BDDAT
016460  100402                            BMI     1$              ;BR IF YES TO "1$".

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

016462  104014                    ERROR   14                      ;ERROR CLOCK B "B OVERFL FLAG" (CSR BIT7)
                                                                  ;FAILED TO SET ON OVERFLOW.

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

```
5413  016464  000411                          BR      2$
5414  016466                          1$:                             ;WAS COUNTER RELOAD AFTER OVERFLOW
5415
5416                                  ;*        MOV     @BCR,$BDDAT   ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
5417  016476  022737  000377  001126           CMP     #377,$BDDAT
5418
5419  016504  001401                          BEQ     2$              ;BR IF YES TO NEXT TEST.
5420
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$

5424  016506  104014                          ERROR   14              ;ERROR CLOCK B COUNT REG. FAILED
5425                                                                  ;TO BE RELOADED AFTER OVERFLOW.
5426
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$

5430  016510                          2$:
5431
5432
5433                                  ;;*****************************************************************
5434                                  ;*TEST 125      *TEST THE INIT. ABILITY OF CLOCK B'S COUNT REG.
5435                                  ;*
5436                                  ;*ALL WE'RE GOING TO DO HERE IS TEST THE INITIALIZE INPUTS
5437                                  ;*TO THE 74193 ICS THAT FORM THE COUNT REGISTER.
5438                                  ;*
5439                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
5440                                  ;*
5441                                  ;*
5442                                  ;*****************************************************************
5443  016510  000004                  TST125: SCOPE
5444  016512  012737  000001  001166          MOV     #1,$TIMES       ;;DO 1 ITERATION
5445
5446                                                                  ;CLEAR CLOCK A.
5447                                                                  ;CLEAR CLOCK B.
5448                                                                  ;LOAD ALL ONES TO BUFFER + COUNT REGS.
5449  016520  005037  001124                  CLR     $GDDAT
5450
5451                                  ;*        MOV     $GDDAT,@ASR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
5452
5453                                  ;*        MOV     $GDDAT,@BSR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
5454  016544  012737  000377  001124           MOV     #377,$GDDAT
5455
5456                                  ;*        MOV     $GDDAT,@BBR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
5457
5458  016562  004737  033650                  JSR     PC,$RESET                       ;DO SYSTEM INIT - GENERATES "INIT" H.
5459
5460  016566  005037  001124                  CLR     $GDDAT          ;FIX $GDDAT FOR ERROR TYPEOUT, IF NEEDED.
5461                                                                  ;READ B'S COUNT REGISTER, INIT
5462
5463                                  ;*        MOV     @BCR,$BDDAT   ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
5464  016602  005737  001126                  TST     $BDDAT
```

```
5465
5466  016606  001401                          BEQ     1$              ;SHOULD HAVE MADE IT AL ZEROS.
5467                                                                  ;BR IF YES - NEXT TEST.
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5471  016610  104007                          ERROR   7               ;ERROR CLOCK B - INIT FAILED TO CLEAR
5472                                                                  ;COUNT REG.
5473
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5477  016612                        1$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 126      *TEST THAT CLOCK B DOESN'T COUNT WHEN NO RATE IS SELECTED
                                        ;;*****************************************************************
016612  000004                        TST126: SCOPE
016614  012737  000100  001166              MOV     #100,$TIMES     ;;DO 100 ITERATIONS

                                                                    ;CLEAR CLOCK A.
                                                                    ;CLEAR CLOCK B.
                                                                    ;ZEROS TO BUFFER + COUNT REGS.
                                                                    ;ENABLE COUNTER TO COUNT - RATE - 0
                                                                    ;(NO RATE).
016622  005037  001124                  CLR     $GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
016656  005237  001124                  INC     $GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
016672  005000                          CLR     R0              ;DELAY FOR ANY COUNT THAT
016674  105200                  1$:     INCB    R0              ;COULD FALSELY OCCUR.
016676  001376                          BNE     1$              ;THIS DELAY APPROX. 369 MS ON A
                                                                ;PDP 11/20.

                                                                ;DID ANY COUNT OCCUR?

                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
016710  005737  001126                  TST     $BDDAT
016714  001401                          BEQ     2$              ;IF NO BR TO NEXT TEST.

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

016716  104014                          ERROR   14              ;ERROR - CLOCK B COUNTED WHEN ENABLED BUT
                                                                ;NO RATE SELECTED. BETTER FIND OUT
                                                                ;WHATS GENERATING "CLOCK B" L PULSES.

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

016720                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 127      *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"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
016720  000004                    TST127: SCOPE
016722  012737  000020  001166            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
                                                                  ;/CLEAR CLOCK A.
016730  005037  001124            CLR     $GDDAT                  ;/CLEAR CLOCK B.
                                                                  ;/CLEAR THE BUFFER + COUNT REGS.
                                                                  ;/SELECT: RATE: 1MHZ ; GO.
                                ;*  MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
                                ;*  MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
                                ;*  MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
016764  012737  000003  001124            MOV     #1!2,$GDDAT
                                ;*  MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
017002  005000                    CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
017004  005200                1$: INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
017006  001376                    BNE     1$              ;/ON A PDP-11/20.
                                                                  ;/DID COUNTER COUNT AT ALL?
                                ;*  MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
017020  005737  001126            TST     $BDDAT
017024  001010                    BNE     2$              ;/BR IF YES - NEXT TEST.
                                ;*  MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
017036  105737  001126            TSTB    $BDDAT
                                                                  ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                  ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
017042  100401                    BMI     2$              ;/BR IF SET - NEXT TEST.

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

017044  104014                    ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                  ;/AT 1MHZ RATE.

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

```
5579  017046                    2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 130      *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"
                                ;*
                                ;*
                                ;;**********************************************************************
017046  000004                  $ST130: SCOPE
017050  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                ;/CLEAR CLOCK A.
017056  005037  001124          CLR     $GDDAT
                                                                ;/CLEAR CLOCK B.
                                                                ;/CLEAR THE BUFFER + COUNT REGS.
                                                                ;/SELECT: RATE: 100KHZ ; GO.

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
017112  012737  000005  001124          MOV     #1!4,$GDDAT

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

017130  005000                          CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
017132  005200                  1$:     INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
017134  001376                          BNE     1$              ;/ON A PDP-11/20.

                                                                ;/DID COUNTER COUNT AT ALL?

                                ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
017146  005737  001126                  TST     $BDDAT
017152  001010                          BNE     2$              ;/BR IF YES - NEXT TEST.

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
017164  105737  001126                  TSTB    $BDDAT
                                                                ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
017170  100401                          BMI     2$              ;/BR IF SET - NEXT TEST.

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

017172  104014                          ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                ;/AT 100KHZ RATE.

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

```
5636  017174                    2$:
```

```
                                        ;/#
                                ;;**********************************************************************
                                ;*TEST 131      *TEST THE ABILITY OF CLOCK B TO COUNT AT 10KHZ PART 1
                                ;*
                                ;*THIS TEST IS DESIGNED TO TEST CLOCK B'S ABILITY TO COUNT
                                ;*IN RATE: 10KHZ        PART1
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT A"
                                ;*
                                ;*
                                ;;**********************************************************************
017174  000004                  TST131: SCOPE
017176  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                ;/CLEAR CLOCK A.
017204  005037  001124          CLR     $GDDAT
                                                                ;/CLEAR CLOCK B.
                                                                ;/CLEAR THE BUFFER + COUNT REGS.
                                                                ;/SELECT: RATE: 10KHZ ; GO.

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
017240  012737  000007  001124          MOV     #1!6,$GDDAT

                                ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

017256  005000                  1$:     CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
017260  005200                          INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
017262  001376                          BNE     1$              ;/ON A PDP-11/20.

                                                                ;/DID COUNTER COUNT AT ALL?

                                ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
017274  005737  001126                  TST     $BDDAT
017300  001010                          BNE     2$              ;/BR IF YES - NEXT TEST.

                                ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
017312  105737  001126                  TSTB    $BDDAT
                                                                ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
017316  100401                          BMI     2$              ;/BR IF SET - NEXT TEST.

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

017320  104014                          ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                ;/AT 10KHZ RATE.

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

```
 5693  017322                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 132      *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"
                                        ;*
                                        ;*
                                        ;;**********************************************************************
017322  000004                          $ST132: SCOPE
017324  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/CLEAR CLOCK A.
017332  005037  001124                        CLR     $GDDAT
                                                                        ;/CLEAR CLOCK B.
                                                                        ;/CLEAR THE BUFFER + COUNT REGS.
                                                                        ;/SELECT: RATE: 1KHZ ; GO.

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
017366  012737  000011  001124                MOV     #1!10,$GDDAT

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

017404  005000                                CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
017406  005200                          1$:   INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
017410  001376                                BNE     1$              ;/ON A PDP-11/20.

                                                                        ;/DID COUNTER COUNT AT ALL?

                                        ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
017422  005737  001126                        TST     $BDDAT
017426  001010                                BNE     2$              ;/BR IF YES - NEXT TEST.

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
017440  105737  001126                        TSTB    $BDDAT
                                                                        ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                        ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
017444  100401                                BMI     2$              ;/BR IF SET - NEXT TEST.

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

017446  104014                                ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                        ;/AT 1KHZ RATE.

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

```
5750  017450                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 133      *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"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
017450  000004                        TST133: SCOPE
017452  012737  000020  001166              MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                    ;/CLEAR CLOCK A.
017460  005037  001124                      CLR     $GDDAT
                                                                    ;/CLEAR CLOCK B.
                                                                    ;/CLEAR THE BUFFER + COUNT REGS.
                                                                    ;/SELECT: RATE: 100HZ ; GO.

                                        ;*  MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*  MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*  MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO dEVICE REG BBR
017514  012737  000011  001124              MOV     #1!11,$GDDAT

                                        ;*  MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

017532  005000                              CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
017534  005200                        1$:   INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
017536  001376                              BNE     1$              ;/ON A PDP-11/20.

                                                                    ;/DID COUNTER COUNT AT ALL?

                                        ;*  MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
017550  005737  001126                      TST     $BDDAT
017554  001010                              BNE     2$              ;/BR IF YES - NEXT TEST.

                                        ;*  MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
017566  105737  001126                      TSTB    $BDDAT
                                                                    ;/COUNT TO OVERFLOW - SO WE'LL SEE IF
                                                                    ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
017572  100401                              BMI     2$              ;/BR IF SET - NEXT TEST.

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

017574  104014                              ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                    ;/AT 100HZ RATE.

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

```
5807  017576                    2$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 134      *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"
                                        ;*
                                        ;*
                                        ;;*****************************************************************
017576  000004                        TST134: SCOPE
017600  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/CLEAR CLOCK A.
017606  005037  001124                        CLR     $GDDAT
                                                                        ;/CLEAR CLOCK B.
                                                                        ;/CLEAR THE BUFFER + COUNT REGS.
                                                                        ;/SELECT: RATE: LINE-FREQ ; GO.

                                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
017642  012737  000017  001124                MOV     #1!16,$GDDAT

                                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

017660  005000                                CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY.
017662  005200                        1$:     INC     R0              ;/THIS DELAY WILL AMOUNT TO APP. 269 MS.
017664  001376                                BNE     1$              ;/ON A PDP-11/20.

                                                                        ;/DID COUNTER COUNT AT ALL?

                                        ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
017676  005737  001126                        TST     $BDDAT
017702  001010                                BNE     2$              ;/BR IF YES - NEXT TEST.

                                        ;*      MOV     @BSR,$BDDAT     ;/READ DEVICE REG BSR,PUT DATA IN $BDDAT.
017714  105737  001126                        TSTB    $BDDAT
                                                                        ;/COUNT TO oVERFLOW - SO WE'LL SEE IF
                                                                        ;/THE OVERFLOW F/F SET BEFORE WE CRY WOLF.
017720  100401                                BMI     2$              ;/BR IF SET - NEXT TEST.

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

017722  104014                                ERROR   14              ;/ERROR CLOCK B - COUNTER FAILED TO COUNT
                                                                        ;/AT LINE-FREQ RATE.

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

```
                                2$:

                                ;;*****************************************************************
                                ;*TEST 135      *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"
                                ;*
                                ;*
                                ;;*****************************************************************
017724  000004                TST135: SCOPE
                                                        ;CLEAR CLOCK A.
                                                        ;CLEAR CLOCK B.
                                                        ;CLEAR A'S BUFFER + COUNT REGISTERS.
                                                        ;PRESET B'S BUFFER + COUNT TO -1 FROM
                                                        ;OVERFLOW.
                                                        ;SELECT: "DISABLE OSC 1 MHZ"; RATE 1 MHZ;
                                                        ;"FEED B TO A"
                                                        ;SET ENABL. MUST BE SET AFTER "FEED B TO A".
                                                        ;ENABLE CLOCK A; MODE 0; RATE 0.
                                                        ;SIMULATE A 1 MHZ PULSE - THIS PULSE
                                                        ;WILL CLOCK CLOCK B'S COUNTER
                                                        ;REGISTER. AN OVERFLOW WILL
                                                        ;OCCUR - THAT OVERFLOW SHOULD
                                                        ;CLOCK CLOCK A'S COUNT REGISTER.
                                                        ;DID CLOCK A'S COUNT REG GET CLOCKED?
017726  005037  001124                CLR     $GDDAT

                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR

                        ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ABR
017762  012737  000377  001124        MOV     #377,$GDDAT

                        ;*      MOV     $GDDAT,@BBR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BBR
020000  012737  004042  001124        MOV     #4042,$GDDAT

                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
020016  005237  001124        INC     $GDDAT

                        ;*      MOV     $GDDAT,@BSR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG BSR
020032  012737  000001  001124        MOV     #1,$GDDAT

                        ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

                        ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
```

```
5916  020060  052737  004000  001124          BIS     #BIT11,$GDDAT
5917
5918                                  ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR
5919
5920                                  ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
5921  020106  005737  001126                  TST     $BDDAT
5922
5923  020112  001001                          BNE     1$              ;IF YES THEN BR NEXT TEST.
5924
          ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5928  020114  104014                          ERROR   14              ;ERROR - UNABLE TO CLOCK CLOCK A'S
5929                                                                  ;COUNT REGISTER WITH THE OVERFLOW
5930                                                                  ;FROM CLOCK B.
5931                                                                  ;THE MOST LOGICAL PLACE TO START
5932                                                                  ;WOULD BE !A CLOCK TIMING".
5933                                                                  ;"FEED B TO A (1)" H + "B OVERFLOW" H
5934                                                                  ;SHOULD BE COMING TOGETHER GOING
5935                                                                  ;INTO THE MUX - BEGING SELECTED AND
5936                                                                  ;COMING OUT OF THE MUX TO BECOME
5937                                                                  ;"CLOCK A" L.
5938
          ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

5942  020116                          1$:
5943                                  .SBTTL  *
5944                                  .SBTTL  * PHASE 6 CLOCK A+B ADVANCE TESTING
5945                                  .SBTTL  *
5946
```

```
                                        ;/#
                                        ;;*****************************************************************
                                        ;*TEST 136      *TEST THAT THE TRAILING EDGE OF STP1 WILL INCR. COUNTER
                                        ;*
                                        ;*IN THIS TEST WE'LL SEE IF WE CATCH THE FIRST COUNT
                                        ;*AFTER STP1 COMES IN AND SETS THE ENABLE F/F ('ST1 ENB COUNTER'
                                        ;*SET).
                                        ;*WHAT WE SHOULD SEE IS THE LEADING EDGE OF ST1 COME
                                        ;*IN AND SET THE ENB F/F AND THE TRAILING EDGE TRIGGER
                                        ;*A 'CLOCK A' PULSE TO INCREMENT THE COUNTER.
                                        ;*WE KNOW FROM A PREVIOUS TEST THAT AN STP1 WILL
                                        ;*COME IN AND SET 'ENABL CNTR A' F/F AND THAT THE
                                        ;*COUNTER WILL INCREMENT, SO WHATS HAPPENING IS WE'RE ACCUALLY
                                        ;*LOOKING AT THE TRAILING EDGE OF STP1 TO SEE IF ITS DOING
                                        ;*THE INCREMENTING
                                        ;*
                                        ;*
                                        ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                        ;*
                                        ;;*****************************************************************
020116  000004                    TST136: SCOPE

                                                                        ;CLR CLK A, SET 'ST1 ENB COUNTER'.;RATE:STP1.
                                                                        ;CLR COUNT + BUFFER REGS.
                                                                        ;GENERATE A MAINTENANCE ST1.
                                                                        ;THE LEADING EDGE SHOULD CAUSE
                                                                        ;'.ENABL CNTR A' F/F TO SET (CSR BIT 00).
                                                                        ;THE TRAILING EDGE SHOULD CLOCK
                                                                        ;THE COUNTER ONCE.
020120  012737  000001  001356            MOV     #BIT0,$TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
020136  005037  001124                  CLR     $GDDAT

                                ;*      MOV     $GDDAT,@ABR     ;/ PUT DATA FROM $GDDAT TO DEViCE REG ABR

                                ;*      MOV     @ASR,$GDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $GDDAT.
020162  052737  010000  001124          BIS     #BIT12,$GDDAT

                                ;*      MOV     $GDDAT,@ASR     ;/ PUT DATA FROM $GDDAT TO DEVICE REG ASR

020200  012737  000001  001124          MOV     #1,$GDDAT       ;SET S/B FOR ERROR TYPEOUT IF NEEDED.
                                                                ;READ THE COUNT REGISTER.

                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
020216  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THE COUNT REG COUNT ONCE?
020224  001401                          BEQ     1$              ;BR IF YES TO NEXT TESt.

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

6000
```

N14


```
 6001  020226  104012                          ERROR   12              ;ST1 FAILED TO COUNT
 6002                                                                  ;CLOCK A'S COUNT REG. AFTER SETTING
 6003                                                                  ;'ENABL CNTR A' - SEE TEST HEADING
 6004                                                                  ;COMMENTS
 6005
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 6009
 6010  020230                            1$:
 6011
 6012
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 137      *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
                                        ;*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"
                                        ;*
                                        ;;*****************************************************************
020230  000004                          $ST137: SCOPE
020232  012737  000020  001166                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/CLEAR CLOCK B.
                                                                        ;/CLEAR CLOCK A.
                                                                        ;/CLEAR A'S BUFFER + COUNT REGS.
                                                                        ;/DISABLE THE 1MHZ OSC.
                                                                        ;/ENABLE CNTR, RATE: 100KHZ ;MODE.

020240  005037  001356                          CLR     $TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                        ;*      MOV     $TMDAT,@ABR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ABR
020274  012737  004000  001356                  MOV     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
020322  052737  000405  001356                  BIS     #401!4,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                                                        ;/SET TO GENERATE ON 1MHZ PULSES
020340  012700  177766                          MOV     #-10.,R0

020344                                  1$:                             ;/GENERATE 1 1MHZ PULSE
                                                                        ;/HAS COUNTER ADVANCED ANY?

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
020354  052737  004000  001356                  BIS     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
```

```
6067  020402  005737  001126                TST     $BDDAT
6068  020406  001002                        BNE     10$             ;/IF SO EXIT THIS LOOP.
6069                                                                ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6070                                                                ;/OSC., THE DIVIDER COULD HAVE
6071                                                                ;/AND COUNT LEFT IN IT.
6072                                                                ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
6073  020410  005200                        INC     R0              ;/DONE 10. 1MHZ PULSES?
6074  020412  001354                        BNE     1$              ;/IF NOT - DO ANOTHER.
6075
6076  020414  012737  000001  001124  10$:  MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6077
6078                                                                ;/READ THE COUNTER.
6079
6080                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6081
6082  020432  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6083  020440  001402                        BEQ     2$              ;/IF YES - NEXT CHECK.
6084
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6088  020442  104012                        ERROR   12              ;/ERROR - CLOCK A - 100KHZ - PULSE
6089                                                                ;/NOT GENERATED WHEN 10 1MHZ PULSES
6090
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6094  020444  000430                        BR      4$
6095  020446  012700  000011          2$:   MOV     #9.,R0  ;/GET THE NUMBER OF '1 MHZ' H PULSES
6096
6097  020452                          3$:                             ;/GENERATE 9 1MHZ PULSES
6098
6099                                ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMdAT.
6100  020462  052737  004000  001356        BIS     #BIT11,$TMDAT
6101
6102                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6103  020500  005300                        DEC     R0              ;/INORDER TO CHECK TO SEE THAT
6104  020502  001363                        BNE     3$              ;/WE DON'T GENERATE ANOTHER 100KHZ PULSE.
6105
6106                                                                ;/READ THE COUNTER
6107
6108                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6109  020514  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100KHZ PULSE GENERATED?
6110  020522  001401                        BEQ     4$              ;/NO-GO TO NEXT TEST!
6111
6112
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6116  020524  104012                        ERROR   12              ;/ERROR CLOCK A WE SEEM TO HAVE
6117                                                                ;/GENERATED A SECOND 100KHZ PULSE
6118                                                                ;/ON ONLY 9 1MHZ PULSES.
6119
              ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
 6123  020526                          4$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 140       *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
                                        ;*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"
                                        ;*
                                        ;;*****************************************************************
020526  000004                        TST140: SCOPE
020530  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/CLEAR CLOCK B.
                                                                        ;/CLEAR CLOCK A.
                                                                        ;/CLEAR A'S BUFFER + COUNT REGS.
                                                                        ;/DISABLE THE 1MHZ OSC.
                                                                        ;/ENABLE CNTR, RATE: 10KHZ ;MODE.

020536  005037  001356                        CLR     $TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                        ;*      MOV     $TMDAT,@ABR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ABR
020572  012737  004000  001356                MOV     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
020620  052737  000407  001356                BIS     #401!6,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                                                        ;/SET TO GENERATE ON 1MHZ PULSES
020636  012700  177634                        MOV     #-100.,R0

020642                                  1$:                             ;/GENERATE 1 1MHZ PULSE
                                                                        ;/HAS COUNTER ADVANCED ANY?

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
020652  052737  004000  001356                BIS     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
```

```
6178  020700  005737  001126                TST     $BDDAT
6179  020704  001002                        BNE     10$             ;/IF SO EXIT THIS LOOP.
6180                                                                ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6181                                                                ;/OSC. THE DIVIDER COULD HAVE
6182                                                                ;/AND COUNT LEFT IN IT.
6183                                                                ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
6184  020706  005200                        INC     R0              ;/DONE 100. 1MHZ PULSES?
6185  020710  001354                        BNE     1$              ;/IF NOT - DO ANOTHER.
6186
6187  020712  012737  000001  001124  10$:  MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6188
6189                                                                ;/READ THE COUNTER.
6190
6191                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6192
6193  020730  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6194  020736  001402                        BEQ     2$              ;/IF YES - NEXT CHECK.
6195
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6199  020740  104012                        ERROR   12              ;/ERROR - CLOCK A - 10KHZ - PULSE
6200                                                                ;/NOT GENERATED WHEN 10 100KHZ PULSES
6201
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6205  020742  000430                        BR      4$
6206  020744  012700  000143          2$:   MOV     #99.,R0 ;/GET THE NUMBER OF '1 MHZ' H PULSES
6207
6208  020750                          3$:                           ;/GENERATE 9 100KHZ PULSES
6209
6210                                ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6211  020760  052737  004000  001356        BIS     #BIT11,$TMDAT
6212
6213                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6214  020776  005300                        DEC     R0              ;/INORDER TO CHECK TO SEE THAT
6215  021000  001363                        BNE     3$              ;/WE DON'T GENERATE ANOTHER 10KHZ PULSE.
6216
6217                                                                ;/READ THE COUNTER
6218
6219                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6220  021012  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 10KHZ PULSE GENERATED?
6221  021020  001401                        BEQ     4$              ;/NO-GO TO NEXT TEST!
6222
6223
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6227  021022  104012                        ERROR   12              ;/ERROR CLOCK A WE SEEM TO HAVE
6228                                                                ;/GENERATED A SECOND 10KHZ PULSE
6229                                                                ;/ON ONLY 9 100KHZ PULSES.
6230
            ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
6234  021024                          4$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 141      *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
                                        ;*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"
                                        ;*
                                        ;;*****************************************************************
021024  000004                          TST141: SCOPE
021026  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/CLEAR CLOCK B.
                                                                        ;/CLEAR CLOCK A.
                                                                        ;/CLEAR A'S BUFFER + COUNT REGS.
                                                                        ;/DISABLE THE 1MHZ OSC.
                                                                        ;/ENABLE CNTR, RATE: 1KHZ ;MODE.

021034  005037  001356                      CLR     $TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                        ;*      MOV     $TMDAT,@ABR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ABR
021070  012737  004000  001356                MOV     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
021116  052737  000411  001356                BIS     #401!10,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                                                        ;/SET TO GENERATE ON 1MHZ PULSES
021134  012700  176030                      MOV     #-1000.,R0

021140                                  1$:                             ;/GENERATE 1 1MHZ PULSE
                                                                        ;/HAS COUNTER ADVANCED ANY?

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
021150  052737  004000  001356                BIS     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEViCE REG ASR

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
```

```
6289  021176  005737  001126            TST     $BDDAT
6290  021202  001002                    BNE     10$             ;/IF SO EXIT THIS LOOP.
6291                                                            ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6292                                                            ;/OSC., THE DIVIDER COULD HAVE
6293                                                            ;/AND COUNT LEFT IN IT.
6294                                                            ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
6295  021204  005200                    INC     R0              ;/DONE 1000. 1MHZ PULSES?
6296  021206  001354                    BNE     1$              ;/IF NOT - DO ANOTHER.
6297
6298  021210  012737  000001  001124  10$:   MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6299
6300                                                            ;/READ THE COUNTER.
6301
6302                            ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6303
6304  021226  023737  001126  001124      CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6305  021234  001402                    BEQ     2$              ;/IF YES - NEXT CHECK.
6306
        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6310  021236  104012                    ERROR   12              ;/ERROR - CLOCK A - 1KHZ - PULSE
6311                                                            ;/NOT GENERATED WHEN 10 10KHZ PULSES
6312
        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6316  021240  000430                    BR      4$
6317  021242  012700  001747      2$:   MOV     #999.,R0        ;/GET THE NUMBER OF '1 MHZ' H PULSES
6318
6319  021246                      3$:                           ;/GENERATE 9 10KHZ PULSES
6320
6321                            ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMdAT.
6322  021256  052737  004000  001356      BIS     #BIT11,$TMDAT
6323
6324                            ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6325  021274  005300                    DEC     R0              ;/INORDER TO CHECK TO SEE THAT
6326  021276  001363                    BNE     3$              ;/WE DON'T GENERATE ANOTHER 1KHZ PULSE.
6327
6328                                                            ;/READ THE COUNTER
6329
6330                            ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6331  021310  023737  001126  001124      CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 1KHZ PULSE GENERATED?
6332  021316  001401                    BEQ     4$              ;/NO-GO TO NEXT TEST!
6333
6334
        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6338  021320  104012                    ERROR   12              ;/ERROR CLOCK A WE SEEM TO HAVE
6339                                                            ;/GENERATED A SECOND 1KHZ PULSE
6340                                                            ;/ON ONLY 9 10KHZ PULSES.
6341
        ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
6345  021322                          4$:
```

```
                                        ;/*
                                        ;;*****************************************************************
                                        ;*TEST 142      *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
                                        ;*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"
                                        ;*
                                        ;*****************************************************************
021322  000004                        TST142: SCOPE
021324  012737  000020  001166                MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                        ;/CLEAR CLOCK B.
                                                                        ;/CLEAR CLOCK A.
                                                                        ;/CLEAR A'S BUFFER + COUNT REGS.
                                                                        ;/DISABLE THE 1MHZ OSC.
                                                                        ;/ENABLE CNTR, RATE: 100HZ ;MODE.

021332  005037  001356                      CLR     $TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMdAT TO DEVICE REG ASR

                                        ;*      MOV     $TMDAT,@ABR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ABR
021366  012737  004000  001356                MOV     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
021414  052737  000413  001356                BIS     #401!12,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMdAT TO DEVICE REG ASR

                                                                        ;/SET TO gENERATE ON 1MHZ PULSES
021432  012700  154360                        MOV     #-10000.,R0

021436                                  1$:                             ;/GENERATE 1 1MHZ PULSE
                                                                        ;/HAS COUNTER ADVANCED ANY?

                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
021446  052737  004000  001356                BIS     #BIT11,$TMDAT

                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                        ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
```

```
6400  021474  005737  001126            TST     $BDDAT
6401  021500  001002                    BNE     10$                ;/IF SO EXIT THIS LOOP.
6402                                                               ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6403                                                               ;/OSC. THE DIVIDER COULD HAVE
6404                                                               ;/AND COUNT LEFT IN IT.
6405                                                               ;/AFTER THIS LOOP ,WE SHOULD BE SUNK.
6406  021502  005200                    INC     R0                 ;/DONE 10000. 1MHZ PULSES?
6407  021504  001354                    BNE     1$                 ;/IF NOT - DO ANOTHER.
6408
6409  021506  012737  000001  001124  10$:   MOV     #1,$GDDAT          ;/SET FOR ERROR TYPEOUT IF NEEDED.
6410
6411                                                               ;/READ THE COUNTER.
6412
6413                            ;*      MOV     @ACR,$BDDAT        ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6414
6415  021524  023737  001126  001124      CMP     $BDDAT,$GDDAT      ;/DID THE COUNTER ADVANCE ONCE?
6416  021532  001402                    BEQ     2$                 ;/IF YES - NEXT CHECK.
6417
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6421  021534  104012                    ERROR   12                 ;/ERROR - CLOCK A - 100HZ - PULSE
6422                                                               ;/NOT GENERATED WHEN 10 1KHZ PULSES
6423
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6427  021536  000430                    BR      4$
6428  021540  012700  023417      2$:   MOV     #9999.,R0          ;/GET THE NUMBER OF '1 MHZ' H PULSES
6429
6430  021544                      3$:                              ;/GENERATE 9 1KHZ PULSES
6431
6432                            ;*      MOV     @ASR,$TMDAT        ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6433  021554  052737  004000  001356      BIS     #BIT11,$TMDAT
6434
6435                            ;*      MOV     $TMDAT,@ASR        ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6436  021572  005300                    DEC     R0                 ;/INORDER TO cHECK TO SEE THAT
6437  021574  001363                    BNE     3$                 ;/WE DON'T GENERATE ANOTHER 100HZ PULSE.
6438
6439                                                               ;/READ THE COUNTER
6440
6441                            ;*      MOV     @ACR,$BDDAT        ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
6442  021606  023737  001126  001124      CMP     $BDDAT,$GDDAT      ;/WAS ANOTHER 100HZ PULSE GENERATED?
6443  021614  001401                    BEQ     4$                 ;/NO-GO TO NEXT TEST!
6444
6445
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6449  021616  104012                    ERROR   12                 ;/ERROR CLOCK A WE SEEM TO HAVE
6450                                                               ;/GENERATED A SECOND 100HZ PULSE
6451                                                               ;/ON ONLY 9 1KHZ PULSES.
6452
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
021620                          4$:
                                ;;*****************************************************************
                                ;*TEST 143      *TEST CLOCK B'S 100KHZ DIVIDER
                                ;*
                                ;+IN THIS TEST WE'LL SEE IF THE 100HZ DIVIDER WILL DIVIDE 1KHZ
                                ;+BY 10 TO GIVE US A 100HZ CLK L PULSE.
                                ;+TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
                                ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100HZ
                                ;*PULSES.
                                ;*THEN WE'LL GENERATE 9 MORE 1KHZ PULSES AND MAKE
                                ;*SURE THAT WE DON'T GET ANOTHER 100HZ PULSE.
                                ;*
                                ;*
                                ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
                                ;*
                                ;*****************************************************************
021620  000004                  TST143: SCOPE
021622  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                ;/CLEAR CLOCK B.
                                                                ;/CLEAR CLOCK A.

                                                                ;/CLEAR B'S BUFFER + COUNT REGS.
                                                                ;/DISABLE THE 1MHZ OSC.
                                                                ;/RATE: 100KHZ
                                                                ;/ENABLE CLOCK B.
021630  005037  001356                  CLR     $TMDAT

                                ;*      MOV     $TMdAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                ;*      MOV     $TMDAT,@BBR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BBR
021664  012737  004040  001356          MOV     #BIT11!BIT5,$TMDAT

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                ;*      MOV     @BSR,$TMDAT     ;/READ DEVICE REG BSR,PUT DATA IN $TMdAT.
021712  052737  000004  001356          BIS     #4,$TMDAT

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMdAT TO DEVICE REG BSR
021730  005237  001356                  INC     $TMDAT

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO dEVICE REG BSR
021744  012700  177766                  MOV     #-10.,R0        ;/SET TO GENERATE 10. 1MHZ PULSES

021750                          1$:                             ;/GENERATE 1 1MHZ PULSE
                                                                ;/HAS THE COUNTER ADVANCED?
```

```
6508                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6509  021760  052737  004000  001356            BIS     #BIT11,$TMDAT
6510
6511                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6512
6513                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6514  022006  005737  001126                    TST     $BDDAT
6515  022012  001002                            BNE     10$             ;/EXIT LOOP IF SO.
6516                                                                    ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6517                                                                    ;/      OSC., THE DIVIDER COULD HAVE
6518                                                                    ;/      HAD ANY COUNT IN IT.
6519                                                                    ;/AFTER THIS LOOP, WE SHOULD BE SUNK.
6520
6521  022014  005200                            INC     R0              ;/DONE 10. 1MHZ PULSES?
6522  022016  001354                            BNE     1$              ;/IF NOT - DO ANOTHER.
6523
6524  022020  012737  000001  001124  10$:     MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6525
6526                                                                    ;/READ THE COUNTER
6527
6528                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6529
6530  022036  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6531  022044  001402                            BEQ     2$              ;/IF YES - NEXT CHECK
6532
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6536  022046  104015                    ERROR   15                      ;/ERROR - CLOCK B - 100KHZ - PULSE
6537                                                                    ;/NOT GENERATED WHEN 10 1MHZ PULSE
6538                                                                    ;/WERE GENERATED.
6539
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6543  022050  000430                            BR      4$
6544
6545  022052  012700  177767          2$:     MOV     #-9.,R0 ;/GET THE NUMBER OF "1 MHZ" H PULSES
6546                                                                    ;/NEED TO GIVE 9 1MHZ PULSES
6547  022056                          3$:
6548
6549                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6550  022066  052737  004000  001356            BIS     #BIT11,$TMDAT
6551
6552                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6553  022104  005200                            INC     R0              ;/IN ORDER TO CHECK TO SEE THAT
6554  022106  001363                            BNE     3$              ;/WE DON'T GENERATE ANOTHER 100KHZ PULSE
6555
6556
6557                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6558  022120  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100KHZ PULSE GENERATED?
6559  022126  001401                            BEQ     4$              ;/NO - GO TO NEXT CHECK.
6560
6561
```

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

 6565  022130  104015                          ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
 6566                                                                  ;/GENERATED A SECOND 100KHZ PULSE
 6567                                                                  ;/ON ONLY 9 1MHZ PULSES.
 6568
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 6572  022132                          4$:
 6573
 6574                                  ;;*********************************************************************
 6575                                  ;*TEST 144      *TEST CLOCK B'S 10KHZ DIVIDER
 6576                                  ;*
 6577                                  ;+IN THIS TEST WE'LL SEE IF THE 100HZ DIVIDER WILL DIVIDE 1KHZ
 6578                                  ;+BY 10 TO GIVE US A 100HZ CLK L PULSE.
 6579                                  ;+TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
 6580                                  ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100HZ
 6581                                  ;*PULSES.
 6582                                  ;*THEN WE'LL GENERATE 9 MORE 1KHZ PULSES AND MAKE
 6583                                  ;*SURE THAT WE DON'T GET ANOTHER 100HZ PULSE.
 6584                                  ;*
 6585                                  ;*
 6586                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF A"
 6587                                  ;*
 6588                                  ;;*********************************************************************
 6589  022132  000004                  TST144: SCOPE
 6590  022134  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
 6591
 6592                                                                  ;/CLEAR CLOCK B.
 6593                                                                  ;/CLEAR CLOCK A.
 6594
 6595                                                                  ;/CLEAR B'S BUFFER + COUNT REGS.
 6596                                                                  ;/DISABLE THE 1MHZ OSC.
 6597                                                                  ;/RATE: 10KHZ
 6598                                                                  ;/ENABLE CLOCK B.
 6599  022142  005037  001356                  CLR     $TMDAT
 6600
 6601                                  ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
 6602
 6603                                  ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
 6604
 6605                                  ;*      MOV     $TMDAT,@BBR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BBR
 6606  022176  012737  004040  001356          MOV     #BIT11!BIT5,$TMDAT
 6607
 6608                                  ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
 6609
 6610                                  ;*      MOV     @BSR,$TMDAT     ;/READ DEVICE REG BSR,PUT DATA IN $TMDAT.
 6611  022224  052737  000006  001356          BIS     #6,$TMDAT
 6612
 6613                                  ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
 6614  022242  005237  001356                  INC     $TMDAT
 6615
```

```
6616                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6617  022256  012700  177634                    MOV     #-100.,R0       ;/SET TO GENERATE 100. 1MHZ PULSES
6618
6619  022262                            1$:                             ;/GENERATE 1 1MHZ PULSE
6620                                                                    ;/HAS THE COUNTER ADVANCED?
6621
6622                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6623  022272  052737  004000  001356              BIS     #BIT11,$TMDAT
6624
6625                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6626
6627                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6628  022320  005737  001126                    TST     $BDDAT
6629  022324  001002                            BNE     10$             ;/EXIT LOOP IF SO.
6630                                                                    ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6631                                                                    ;/      OSC., THE DIVIDER COULD HAVE
6632                                                                    ;/      HAD ANY COUNT IN IT.
6633                                                                    ;/AFTER THIS LOOP, WE SHOULD BE SUNK.
6634
6635  022326  005200                            INC     R0              ;/DONE 100. 1MHZ PULSES?
6636  022330  001354                            BNE     1$              ;/IF NOT - DO ANOTHER.
6637
6638  022332  012737  000001  001124  10$:     MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6639
6640                                                                    ;/READ THE COUNTER
6641
6642                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6643
6644  022350  023737  001126  001124              CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6645  022356  001402                            BEQ     2$              ;/IF YES - NEXT CHECK
6646
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6650  022360  104015                    ERROR   15                      ;/ERROR - CLOCK B - 10KHZ - PULSE
6651                                                                    ;/NOT GENERATED WHEN 10 100KHZ PULSE
6652                                                                    ;/WERE GENERATED.
6653
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6657  022362  000430                    BR      4$
6658
6659  022364  012700  177635    2$:     MOV     #-99.,R0                ;/GET THE NUMBER OF "1 MHZ" H PULSES
6660                                                                    ;/NEED TO GIVE 9 100KHZ PULSES
6661  022370                            3$:
6662
6663                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6664  022400  052737  004000  001356              BIS     #BIT11,$TMDAT
6665
6666                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6667  022416  005200                            INC     R0              ;/IN ORDER TO CHECK TO SEE THAT
6668  022420  001363                            BNE     3$              ;/WE DON'T GENERATE ANOTHER 10KHZ PULSE
6669
```

```
                                ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
022432  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 10KHZ PULSE GENERATED?
022440  001401                        BEQ     4$              ;/NO - GO TO NEXT CHECK.

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

022442  104015                        ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
                                                                ;/GENERATED A SECOND 10KHZ PULSE
                                                                ;/ON ONLY 9 100KHZ PULSES.

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

022444                          4$:

                                ;*************************************************************
                                ;*TEST 145      *TEST CLOCK B'S 1KHZ 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
                                ;*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"
                                ;*
                                ;*************************************************************
022444  000004                  TST145: SCOPE
022446  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS

                                                                ;/CLEAR CLOCK B.
                                                                ;/CLEAR CLOCK A.

                                                                ;/CLEAR B'S BUFFER + COUNT REGS.
                                                                ;/DISABLE THE 1MHZ OSC.
                                                                ;/RATE: 1KHZ
                                                                ;/ENABLE CLOCK B.
022454  005037  001356                CLR     $TMDAT

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                ;*      MOV     $TMDAT,@BBR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BBR
022510  012737  004040  001356          MOV     #BIT11!BIT5,$TMDAT

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
```

```
6724                                    ;*      MOV     @BSR,$TMDAT     ;/READ DEVICE REG BSR,PUT DATA IN $TMDAT.
6725  022536  052737  000010  001356              BIS     #10,$TMDAT
6726
6727                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6728  022554  005237  001356                      INC     $TMDAT
6729
6730                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6731  022570  012700  176030                      MOV     #-1000.,R0      ;/SET TO GENERATE 1000. 1MHZ PULSES
6732
6733  022574                          1$:                             ;/GENERATE 1 1MHZ PULSE
6734                                                                  ;/HAS THE COUNTER ADVANCED?
6735
6736                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6737  022604  052737  004000  001356              BIS     #BIT11,$TMDAT
6738
6739                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6740
6741                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6742  022632  005737  001126                      TST     $BDDAT
6743  022636  001002                              BNE     10$             ;/EXIT LOOP IF SO.
6744                                                                  ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6745                                                                  ;/      OSC., THE DIVIDER COULD HAVE
6746                                                                  ;/      HAD ANY COUNT IN IT.
6747                                                                  ;/AFTER THIS LOOP, WE SHOULD BE SUNK.
6748
6749  022640  005200                              INC     R0              ;/DONE 1000. 1MHZ PULSES?
6750  022642  001354                              BNE     1$              ;/IF NOT - DO ANOTHER.
6751
6752  022644  012737  000001  001124  10$:       MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6753
6754                                                                  ;/READ THE COUNTER
6755
6756                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6757
6758  022662  023737  001126  001124              CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6759  022670  001402                              BEQ     2$              ;/IF YES - NEXT CHECK
6760
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6764  022672  104015                          ERROR   15                  ;/ERROR - CLOCK B - 1KHZ - PULSE
6765                                                                  ;/NOT GENERATED WHEN 10 10KHZ PULSE
6766                                                                  ;/WERE GENERATED.
6767
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6771  022674  000430                              BR      4$
6772
6773  022676  012700  176031          2$:       MOV     #-999.,R0       ;/GET THE NUMBER OF "1 MHZ" H PULSES
6774                                                                  ;/NEED TO GIVE 9 10KHZ PULSES
6775  022702                          3$:
6776
6777                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
```

```
6778  022712  052737  004000  001356          BIS     #BIT11,$TMDAT
6779
6780                                  ;*        MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6781  022730  005200                          INC     R0              ;/IN ORDER TO CHECK TO SEE THAT
6782  022732  001363                          BNE     3$              ;/WE DON'T GENERATE ANOTHER 1KHZ PULSE
6783
6784
6785                                  ;*        MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6786  022744  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 1KHZ PULSE GENERATED?
6787  022752  001401                          BEQ     4$              ;/NO - GO TO NEXT CHECK.
6788
6789
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6793  022754  104015                          ERROR   15              ;/ERROR CLOCK B WE SEEM TO HAVE
6794                                                                  ;/GENERATED A SECOND 1KHZ PULSE
6795                                                                  ;/ON ONLY 9 10KHZ PULSES.
6796
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

6800  022756                        4$:
6801
6802                                  ;;*****************************************************************
6803                                  ;*TEST 146      *TEST CLOCK B'S 100HZ DIVIDER
6804                                  ;*
6805                                  ;+IN THIS TEST WE'LL SEE IF THE 100HZ DIVIDER WILL DIVIDE 1KHZ
6806                                  ;+BY 10 TO GIVE US A 100HZ CLK L PULSE.
6807                                  ;+TO DO THIS, WE'LL DISABLE THE REGULAR 1MHZ CLK PULSE AND
6808                                  ;*PULSES THAT IN TURN SHOULD DIVIDE BY TEN TO GIVE US ONE 100HZ
6809                                  ;*PULSES.
6810                                  ;*THEN WE'LL GENERATE 9 MORE 1KHZ PULSES AND MAKE
6811                                  ;*SURE THAT WE DON'T GET ANOTHER 100HZ PULSE.
6812                                  ;*
6813                                  ;*
6814                                  ;* PROBABLE SYNC POINT FOR THIS TESt:: "LD BUFF A"
6815                                  ;*
6816                                  ;*****************************************************************
6817  022756  000004          TST146: SCOPE
6818  022760  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
6819
6820                                                                  ;/CLEAR CLOCK B.
6821                                                                  ;/CLEAR CLOCK A.
6822
6823                                                                  ;/CLEAR B'S BUFFER + COUNT REGS.
6824                                                                  ;/DISABLE THE 1MHZ OSC.
6825                                                                  ;/RATE: 100HZ
6826                                                                  ;/ENABLE CLOCK B.
6827  022766  005037  001356                  CLR     $TMDAT
6828
6829                                  ;*        MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6830
6831                                  ;*        MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEViCE REG ASR
```

```
6832
6833                                    ;*      MOV     $TMDAT,@BBR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BBR
6834  023022  012737  004040  001356            MOV     #BIT11!BIT5,$TMDAT
6835
6836                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6837
6838                                    ;*      MOV     @BSR,$TMDAT     ;/READ DEVICE REG BSR,PUT DATA IN $TMDAT.
6839  023050  052737  000012  001356            BIS     #12,$TMDAT
6840
6841                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6842  023066  005237  001356                    INC     $TMDAT
6843
6844                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
6845  023102  012700  154360                    MOV     #-10000.,R0     ;/SET TO GENERATE 10000. 1MHZ PULSES
6846
6847  023106                            1$:                             ;/GENERATE 1 1MHZ PULSE
6848                                                                    ;/HAS THE COUNTER ADVANCED?
6849
6850                                    ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
6851  023116  052737  004000  001356            BIS     #BIT11,$TMDAT
6852
6853                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
6854
6855                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6856  023144  005737  001126                    TST     $BDDAT
6857  023150  001002                            BNE     10$             ;/EXIT LOOP IF SO.
6858                                                                    ;/NOTE: WHEN WE DISABLED THE 1 MHZ.
6859                                                                    ;/      OSC., THE DIVIDER COULD HAVE
6860                                                                    ;/      HAD ANY COUNT IN IT.
6861                                                                    ;/AFTER THIS LOOP, WE SHOULD BE SUNK.
6862
6863  023152  005200                            INC     R0              ;/DONE 10000. 1MHZ PULSES?
6864  023154  001354                            BNE     1$              ;/IF NOT - DO ANOTHER.
6865
6866  023156  012737  000001  001124  10$:     MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT IF NEEDED.
6867
6868                                                                    ;/READ THE COUNTER
6869
6870                                    ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
6871
6872  023174  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;/DID THE COUNTER ADVANCE ONCE?
6873  023202  001402                            BEQ     2$              ;/IF YES - NEXT CHECK
6874
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

6878  023204  104015                            ERROR   15              ;/ERROR - CLOCK B - 100HZ - PULSE
6879                                                                    ;/NOT GENERATED WHEN 10 1KHZ PULSE
6880                                                                    ;/WERE GENERATED.
6881
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

6885  023206  000430                            BR      4$
```

```
                                2$:     MOV     #-9999.,R0      ;/GET THE NUMBER OF "1 MHZ" H PULSES
                                                                ;/NEED TO GIVE 9 1KHZ PULSES
                                3$:
                                ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
                                        BIS     #BIT11,$TMDAT
                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
                                        INC     R0              ;/IN ORDER TO CHECK TO SEE THAT
                                        BNE     3$              ;/WE DON'T GENERATE ANOTHER 100HZ PULSE
                                ;*      MOV     @BCR,$BDDAT     ;/READ DEVICE REG BCR,PUT DATA IN $BDDAT.
                                        CMP     $BDDAT,$GDDAT   ;/WAS ANOTHER 100HZ PULSE GENERATED?
                                        BEQ     4$              ;/NO - GO TO NEXT CHECK.

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

                                ERROR   15                      ;/ERROR CLOCK B WE SEEM TO HAVE
                                                                ;/GENERATED A SECOND 100HZ PULSE
                                                                ;/ON ONLY 9 1KHZ PULSES.

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

                                4$:

                                        .SBTTL

                                .SBTTL  END OF PASS ROUTINE

                                ;;**********************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS"
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO LOOP
                                ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
                                ;*$ENDMG CAN BE CHANGED TO 7.

                                $EOP:
                                        NOP
                                        CLR     $TSTNM          ;;ZERO THE TEST NUMBER
                                        CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
                                        INC     $PASS           ;;INCREMENT THE PASS NUMBER
                                        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
                                        DEC     (PC)+           ;;LOOP?
                                $EOPCT: .WORD   1
                                        BGT     $DOAGN          ;;YES
                                        MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
                                $ENDCT: .WORD   1
```

```
                                        $EOpCT
                                        TYPE    ,$ENDMG         ;;TYPE "END PASS"
$GEt42: MOV     @#42,R0                 ;;GET MONITOR ADDRESS
        BEQ     $DOAGN                  ;;BRANCH IF NO MONITOR
        RESET                           ;;CLEAR THE WORLD
$ENDAD: JSR     PC,(R0)                 ;;GO TO MONITOR
        NOP                             ;;SAVE ROOM
        NOP                             ;;FOR
        NOP                             ;;ACT11
$DOAGN:
        JMP     @(PC)+                  ;;RETURN
$RTNAD: .WORD   LOOP
$ENuLL: .BYTE   -1,-1,0                 ;;NULL CHARACTER STRING
$ENDMG: .ASCIZ  <15><12>/END PASS/

        .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 DD AND LL OF J1 TOGETHER.
        ;*
        ;* LOGIC TEST (L + S 200) SHOULD BE RUN FIRST.
        ;*

LS210:

1$:     CLR     $ICNT                   ;/CLEAR ITERATION COUNT
        MOV     #-60.,$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.
```

```
                                                                ;/CLEAR CLOCK A.
                                        2$:                     ;/CLEAR CLOCK B.
023410  005037  001356                        CLR     $TMDAT
                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
                                        ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
                                                                ;GENERATE AN "STP2 OUT" PULSE
                                                                ;AT THIS POINT YOU SHOULD
                                                                ;SEE AN OUTPUT AT PIN V.
                                                                ;PIN V SHOULD BE WIRED TO
                                                                ;PIN LL FOR "SCHMITT TRIG 1" IN.
                                                                ;IS "ST1 FLAG" BIT 15 IN CLOCK
                                                                ;A'S CSR SET?
                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
                                        ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
023454  052737  002000  001356                BIS     #BIT10,$TMDAT
                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
023472  032737  100000  001356                BIT     #BIT15,$TMDAT
                                        ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
023510  001001                                BNE     3$              ;BR IF YES TO 3$:

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

023512  104000                                ERROR                   ;ERROR "SCHMITT TRIG 1" IN NOT
                                                                        ;RECEIVED. HAVE YOU WIRED IT RIGHT?

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

023514                                  3$:
023514  032777  020000  155416                BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
023522  001332                                BNE     2$              ;/YES - IGNORE ANY UPDATES.
023524  005237  001104                        INC     $ICNT           ;/UPDATE COUNT.
023530  001327                                BNE     2$              ;/IF NOT DONE 65,324 TIMES,
                                                                        ;/DO IT AGAIN.
023532  104401  023540                        TYPE    ,65$            ;;TYPE ASCIZ STRING
023536  000401                                BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  #*#
023542                                  64$:
```

```
7048  023542  005237  001210                    INC     $PASS           ;/DONE 60 PASSES?
7049  023546  100720                            BMI     2$              ;/NO - NO NEED FOR CR,LF.
7050  023550  104401  023556                    TYPE    ,67$            ;;TYPE ASCIZ STRING
7051  023554  000402                            BR      66$             ;;GET OVER THE ASCIZ
7052                                  ;;67$:    .ASCIZ  <15><12>##
7053  023562                          66$:
7054  023562  000705                            BR      1$
7055
7056
7057                                  .SBTTL    ;*      "STP1 OUT" TO "SCHMITT TRIG 2" H TESTS
7058                                            ;*
7059                                            ;* THIS IS A SPECIAL TES! SECTION DEVOTED FOR TES!ING AND
7060                                            ;* PROVIDING SCOPE LOOP CAPABILITIES FOR "STP1 OUT" AND
7061                                            ;* "SCHMITT TRIG2" IN.
7062                                            ;*
7063                                            ;* WHEN YOU LOAD AND START AT LOCATION 214, PROGRAM
7064                                            ;* CONTROL IS TRANSFERRED HERE. "STP1 OUT" L PULSES ARE
7065                                            ;* GENERATED BY "LD STAT A HI" + "BD12" H (MAIN ST1).
7066                                            ;* PIN DD ("STP1 OUT") IS WIRED TO PIN BB ("SCHMITT
7067                                            ;* TRIG 2") FOR THIS TEST. "STP1 OUT" PULSES ARE RECEIVED AS
7068                                            ;* "SCHMITT TRIG 2" PULSES WHICH WILL CLEAR CLOCK A'S
7069                                            ;* COUNT REGISTER IF MODE 3 IS SELECTED.
7070                                            ;* IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIC.
7071                                            ;* AND ERROR SWITCH REGISTER OPTIONS ARE USED.
7072                                            ;* AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH
7073                                            ;* THE TEST. SW13=1 WILL INHIBIT THIS FEATURE.
7074                                            ;*
7075                                  ;*
7076                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
7077                                  ;*
7078                                            ;* YOU MUST WIRE PINS DD AND BB OF J1 TOGETHER.
7079                                            ;*
7080                                            ;* LOGIC TESTS (L + S AT 200) SHOULD BE RUN FIRST.
7081                                            ;*
7082
7083  023564                          LS214:
7084
7085  023564  005037  001104          1$:       CLR     $ICNT           ;/CLEAR ITERATION COUNT
7086  023570  012737  177704  001210            MOV     #-60.,$PASS     ;/SET PASS COUNT.
7087                                                                    ;/NOTE: PASS COUNT USED ONLY
7088                                                                    ;/      TO DETECT 60 PASSES SO
7089                                                                    ;/      IT CAN GENERATE A CRLF.
7090                                                                    ;/      AFTER CRLF IT WILL BE ZEROED.
7091
7092  023576                          2$:                               ;/CLEAR CLOCK A.
7093  023576  005037  001356                    CLR     $TMDAT          ;/CLEAR CLOCK B.
7094
7095                                  ;*        MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
7096
7097                                  ;*        MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
7098
7099                                                                    ;LOAD ALL ONES TO CLK A'S BUFFER + COUNT REG.
7100                                                                    ;SELECT MODE 3.
7101                                                                    ;GENERATE A "STP1 OUT" PULSE.
```

```
                                                                ;AT THIS POINT WE SHOULD SEE AN
                                                                ;OUTPUT AT PIN DD.  PIN DD SHOULD
                                                                ;BE WIRED TO PIN BB FOR
                                                                ;"SCHMITT TRIG 2" IN. THIS SHOULD
                                                                ;CAUSE AN "STP2" WHICH WILL CLEAR
                                                                ;CLOCK A'S COUNT REGISTER.
023622  012737  177777  001356          MOV     #-1,$TMDAT

                                ;*      MOV     $TMDAT,@ABR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ABR
023640  005037  001356                  CLR     $TMDAT
023644  012737  001400  001356          MOV     #BIT8!BIT9,$TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
023672  052737  010000  001356          BIS     #BIT12,$TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
                                                                ;WAS THE COUNT REGISTER CLEARED?

                                ;*      MOV     @ACR,$BDDAT     ;/READ DEVICE REG ACR,PUT DATA IN $BDDAT.
023720  005737  001126                  TST     $BDDAT

023724  001401                          BEQ     3$              ;IF YES BR 3$.

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

023726  104000                          ERROR                   ;ERROR "SCHMITT TRIG 2" IN NOT
                                                                ;RECEIVED. HAVE YOU WIRED IT RIGHT?

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

023730                          3$:

023730  032777  020000  155202          BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
023736  001317                          BNE     2$              ;/YES - IGNORE ANY UPDATES.

023740  005237  001104                  INC     $ICNT           ;/UPDATE COUNT.
023744  001314                          BNE     2$              ;/IF NOT DONe 65,324 TIMES,
                                                                ;/DO IT AGAIN.

023746  104401  023754                  TYPE    ,65$            ;;TYPE ASCIZ STRING
023752  000401                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  #**#
023756                          64$:

023756  005237  001210                  INC     $PASS           ;/DONE 60 PASSES?
023762  100705                          BMI     2$              ;/NO - NO NEED FOR CR,LF.
023764  104401  023772                  TYPE    ,67$            ;;TYPE ASCIZ STRING
023770  000402                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>##
023776                          66$:
```

```
7156  023776  000672                          BR      1$
7157
7158
7159                                  .SBTTL  *       "SCHMITT TRIG 3" IN, "ST3 OUT" TESTS
7160                                          ;*
7161                                          ;*THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND
7162                                          ;*PROVIDING SCOPE LOOP CAPABILITIES FOR "SCHMITT TRIG 3"
7163                                          ;*AND "ST3 OUT".
7164                                          ;*
7165                                          ;*WHEN YOU LOAD AND START AT LOCATION 220, PROGRAM
7166                                          ;*CONTROL IS TRANSFERRED HERE. "STP2" PULSES ARE GENERATED
7167                                          ;*BY "LD STAT A H," + "BD10" H (MAIN STP2). PIN V ("STP2 OUT")
7168                                          ;*IS WIRED TO PIN T ("SCHMITT TRIG 3"). "SCHMITT TRIG 3" PULSES
7169                                          ;*GIVE US "ST3 OUT" PULSES. PIN L ("ST3 OUT") IS WIRED
7170                                          ;*TO PIN LL ("SCHMITT TRIG 1"). AND "SCHMITT TRIG 1" WILL SET
7171                                          ;*CLOCK A'S STATUS REGISTER BIT 15.
7172                                          ;*IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIC.
7173                                          ;*AND ERROR SWITCH REGISTER OPTIONS ARE USED.
7174                                          ;*AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE
7175                                          ;*TEST. SW13=1 WILL INHIBIT THIS FEATURE.
7176                                          ;*
7177                                  ;*
7178                                  ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD STAT B"
7179                                  ;*
7180                                          ;*YOU MUST WIRE PINS DD TO T OF J1 TOGETHER, AS WELL AS
7181                                          ;*PINS L TO BB OF J1 TOGETHER.
7182                                          ;*
7183                                          ;*TESTS LS210 AND LS214 SHOULD BE RUN FIRST.
7184                                          ;*
7185
7186  024000                          LS220:
7187
7188  024000  005037  001104          1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
7189  024004  012737  177704  001210          MOV     #-60.,$PASS     ;/SET PASS COUNT.
7190                                                                  ;/NOTE: PASS COUNT USED ONLY
7191                                                                  ;/      TO DETECT 60 PASSES SO
7192                                                                  ;/      IT CAN GENERATE A CRLF.
7193                                                                  ;/      AFTER CRLF IT WILL BE ZEROED.
7194                                                                  ;/CLEAR CLOCK A.
7195  024012                          2$:                             ;/CLEAR CLOCK B.
7196  024012  005037  001356                  CLR     $TMDAT
7197
7198                                  ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
7199
7200                                  ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
7201
7202                                                                  ;GENERATE A "STP2 OUT" PULSE.
7203  024036  005037  001356                  CLR     $TMDAT
7204
7205                                  ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
7206  024052  052737  011000  001356          BIS     #BIT12!BIT9,$TMDAT
7207
7208                                  ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
7209                                                                  ;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.

                                                        ;DID BIT 15 SET?

                                ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
024100  105737  001126                  TSTB    $BDDAT
024104  100401                          BMI     3$              ;IF YES, BR TO 3$.

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

024106  104000                          ERROR                   ;ERROR "SCHMITT TRIG 1" NOT RECEIVED
                                                                ;HAVE YOU WIRED IT RIGHT?

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

024110                          3$:

024110  032777  020000  155022          BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
024116  001335                          BNE     2$              ;/YES - IGNORE ANY UPDATES.

024120  005237  001104                  INC     $ICNT           ;/UPDATE COUNT.
024124  001332                          BNE     2$              ;/IF NOT DONE 65,324 TIMES,
                                                                ;/DO IT AGAIN.

024126  104401  024134                  TYPE    ,65$            ;;TYPE ASCIZ STRING
024132  000401                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  #**#
                                64$:

024136                                  INC     $PASS           ;/DONE 60 PASSES?
024136  005237  001210                  BMI     2$              ;/NO - NO NEED FOR CR,LF.
024142  100723                          TYPE    ,67$            ;;TYPE ASCIZ STRING
024144  104401  024152                  BR      66$             ;;GET OVER THE ASCIZ
024150  000402                  ;;67$:  .ASCIZ  <15><12>##
                                66$:
024156                                  BR      1$
024156  000710


                                .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 BB OF J1 TOGETHER.
                                        ;*
                                        ;*TEST LS210 SHOULD BE RUN FIRST.
                                        ;*
024160                          LS224:

024160  005037  001104          1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
024164  012737  177704  001210          MOV     #-60.,$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.
                                                                ;/CLEAR CLOCK A.
024172                          2$:                             ;/CLEAR CLOCK B.
024172  005037  001356                  CLR     $TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

                                ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR

                                                                ;PRELOAD CLOCK A'S BUFFER + COUNT REGS.
024216  012737  177777  001356          MOV     #-1,$TMDAT

                                ;*      MOV     $TMDAT,@ABR     ;/ PUT DATA FROM $TMdAT TO DEVICE REG ABR

                                                                ;RATE: 1MHZ, ENABLE COUNTER
024234  012737  001003  001356          MOV     #1003,$TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR

024252                          3$:                             ;HAS AN OVERFLOW OCCURRED?

                                ;*      MOV     @ASR,$TMDAT     ;/READ DEVICE REG ASR,PUT DATA IN $TMDAT.
024262  032737  000040  001356          BIT     #BIT05,$TMDAT

                                ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
024300  001764                  ;NO - THEN WAIT FOR IT.
                                ;OVERFLOW SHOULD GO OUT AS
                                ;"A EVENT OUT". YOU SHOULD SEE
                                ;AN OUTPUT AT PIN VV.
                                ;THIS IN TURN IS BROUGHT
                                ;IN AS "SCHMITT TRIG 1" IN TO
                                ;SET CLOCK A'S CSR BIT 15.
```

```
7318
7319                                                                    ;DID CSR BIT 15 SET?
7320
7321                                    ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
7322  024312  105737  001126                    TSTB    $BDDAT
7323  024316  100401                            BMI     4$              ;BR IF YES TO 4$
7324
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

7328  024320  104000                            ERROR                   ;ERROR "A EVENT OUT" PULSE
7329                                                                    ;NOT DETECTED. HAVE YOU
7330                                                                    ;WIRED IT RIGHT?
7331
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$

7335  024322                            4$:
7336
7337  024322  032777  020000  154610            BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
7338  024330  001320                            BNE     2$              ;/YES - IGNORE ANY UPDATES.
7339
7340  024332  005237  001104                    INC     $ICNT           ;/UPDATE COUNT.
7341  024336  001315                            BNE     2$              ;/IF NOT DONE 65,324 TIMES,
7342                                                                    ;/DO IT AGAIN.
7343
7344  024340  104401  024346                    TYPE    ,65$            ;;TYPE ASCIZ STRING
7345  024344  000401                            BR      64$             ;;GET OVER THE ASCIZ
7346                                    ;;65$:  .ASCIZ  #**
7347  024350                            64$:
7348
7349  024350  005237  001210                    INC     $PASS           ;/DONE 60 PASSES?
7350  024354  100706                            BMI     2$              ;/NO - NO NEED FOR CR,LF.
7351  024356  104401  024364                    TYPE    ,67$            ;;TYPE ASCIZ STRING
7352  024362  000402                            BR      66$             ;;GET OVER THE ASCIZ
7353                                    ;;67$:  .ASCIZ  <15><12>**
7354  024370                            66$:
7355  024370  000673                            BR      1$
7356
7357
7358                                    .SBTTL  *       "B EVENT OUT" TEST
7359                                            ;*
7360                                            ;*THIS IS A SPECIAL SECTION DEVOTED FOR TESTING AND
7361                                            ;*PROVIDING SCOPE LOOP CAPABILITIES FOR "B EVENT OUT".
7362                                            ;*
7363                                            ;*WHEN YOU LOAD AND START AT LOCATION 230, PROGRAM
7364                                            ;*CONTROL IS TRANSFERRED HERE. "B EVENT OUT" PULSES ARE
7365                                            ;*GENERATED BY CLOCK B OVERFLOWS. PIN TT
7366                                            ;*("B EVENT OUT") IS WIRED TO PIN LL ("SCHMITT TRIG 1")
7367                                            ;*"SCHMITT TRIG 1" PULSES WILL SET CLOCK A'S CSR BIT 15.
7368                                            ;*IF AN ERROR IS DETECTED, NORMAL ERROR REPORTING TECHNIQ.
7369                                            ;*AND ERROR SWITCH REGISTER OPTIONS ARE USED.
7370                                            ;*AN "*" IS TYPED AFTER EACH 65,324 LOOPS THROUGH THE TEST.
7371                                            ;*SW13=1 WILL INHIBIT THIS FEATURE.
```

```
7372                                            ;*
7373                                    ;*
7374                                    ;* PROBABLE SYNC POINT FOR THIS TEST:: "LD BUFF B"
7375                                    ;*
7376                                            ;*YOU MUST WIRE PINS TT AND BB OF J1 TOGETHER.
7377                                            ;*
7378                                            ;*TEST LS21G SHOULD BE RUN FIRST.
7379                                            ;*
7380
7381  024372                            LS230:
7382
7383  024372  005037  001104            1$:     CLR     $ICNT           ;/CLEAR ITERATION COUNT
7384  024376  012737  177704  001210            MOV     #-60.,$PASS     ;/SET PASS COUNT.
7385                                                                    ;/NOTE: PASS COUNT USED ONLY
7386                                                                    ;/      TO DETECT 60 PASSES SO
7387                                                                    ;/      IT CAN GENERATE A CRLF.
7388                                                                    ;/      AFTER CRLF IT WILL BE ZEROED.
7389                                                                    ;/CLEAR CLOCK A.
7390  024404                            2$:                             ;/CLEAR CLOCK B.
7391  024404  005037  001356                    CLR     $TMDAT
7392
7393                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
7394
7395                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMdAT TO DEVICE REG BSR
7396
7397                                                                    ;PRELOAD CLOCK B'S BUFFER + COUNT REGS.
7398  024430  012737  177777  001356            MOV     #-1,$TMDAT
7399
7400                                    ;*      MOV     $TMDAT,@BBR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BBR
7401  024446  012737  001000  001356            MOV     #1000,$TMDAT
7402
7403                                    ;*      MOV     $TMDAT,@ASR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG ASR
7404
7405                                                                    ;RATE: 1MHZ, ENABLE COUNTER
7406
7407                                                                    ;HAS AN OVERFLOW OCCURRED?
7408  024464  012737  000003  001356            MOV     #3,$TMDAT
7409
7410                                    ;*      MOV     $TMDAT,@BSR     ;/ PUT DATA FROM $TMDAT TO DEVICE REG BSR
7411  024502                            3$:
7412
7413                                    ;*      MOV     @BSR,$TMDAT     ;/READ DEVICE REG BSR,PUT DATA IN $TMDAT.
7414  024512  032737  000200  001356            BIT     #BIT07,$TMDAT
7415  024520  001770                            BEQ     3$              ;NO -THEN WAIT FOR IT.
7416                                                                    ;OVERFLOW SHOULD GO OUT AS
7417                                                                    ;"B EVENT OUT". YOU SHOULD SEE
7418                                                                    ;AN OUTPUT AT PIN TT.
7419                                                                    ;THIS IN TURN IS BROUGHT
7420                                                                    ;IN AS "SCHMITT TRIG 1" IN TO
7421                                                                    ;SET CLOCK A'S CSR BIT 15.
7422
7423                                                                    ;DID CSR BIT 15 SET?
7424
7425                                    ;*      MOV     @ASR,$BDDAT     ;/READ DEVICE REG ASR,PUT DATA IN $BDDAT.
```

```
7426  024532  105737  001126                  TSTB    $BDDAT
7427  024536  100401                          BMI     4$              ;BR IF YES TO 4$
7428
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7432  024540  104000                          ERROR                   ;ERROR "B EVENT OUT" PULSE
7433                                                                  ;NOT DETECTED. HAVE YOU
7434                                                                  ;WIRED IT RIGHT?
7435
                ;;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>> ERROR <<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

7439  024542                        4$:
7440
7441  024542  032777  020000  154370          BIT     #BIT13,@SWR     ;/INHIBIT "*" TYPEOUT?
7442  024550  001315                          BNE     2$              ;/YES - IGNORE ANY UPDATES.
7443
7444  024552  005237  001104                  INC     $ICNT           ;/UPDATE COUNT.
7445  024556  001312                          BNE     2$              ;/IF NOT DONE 65,324 TIMES,
7446                                                                  ;/DO IT AGAIN.
7447
7448  024560  104401  024566                  TYPE    ,65$            ;;TYPE ASCIZ STRING
7449  024564  000401                          BR      64$             ;;GET OVER THE ASCIZ
7450                                ;;65$:    .ASCIZ  #**#
7451  024570                        64$:
7452
7453  024570  005237  001210                  INC     $PASS           ;/DONE 60 PASSES?
7454  024574  100703                          BMI     2$              ;/NO - NO NEED FOR CR,LF.
7455  024576  104401  024604                  TYPE    ,67$            ;;TYPE ASCIZ STRING
7456  024602  000402                          BR      66$             ;;GET OVER THE ASCIZ
7457                                ;;67$:    .ASCIZ  <15><12>##
7458  024610                        66$:
7459  024610  000670                          BR      1$
7460
7461
7462                                ;*ROUTINE TO HANDLE TRAPS TO 10C 4, 10 AND .
7463                                ;*INTERRUPTS TO WRONG VECTORS.
7464                                ;*.+2, IOTT(TRAPS) WERE PUT IN LOCATIONS 4-1000
7465
7466
7467  024612                        IOTRD:
7468  024612  011637  024762                  MOV     (R6),2$         ;GET WHERE WE TRAPPED TO.
7469  024616  162737  000004  024762          SUB     #4,2$           ;=WHERE R6 RETURN 10-4
7470  024624  104401  024632                  TYPE    ,65$            ;;TYPE ASCIZ STRING
7471  024630  000412                          BR      64$             ;;GET OVER THE ASCIZ
7472                                ;;65$:    .ASCIZ  <15><12>#ILLEGAL TRAP TO: #
7473  024656                        64$:
7474
7475  024656  013746  024762                  MOV     2$,-(SP)        ;;SAVE 2$ FOR TYPEOUT
7476  024662  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7477
7478  024664  104401  024672                  TYPE    ,67$            ;;TYPE ASCIZ STRING
7479  024670  000407                          BR      66$             ;;GET OVER THE ASCIZ
```

```
7480                                    ;;67$:  .ASCIZ  # FROM LOC.: #
7481  024710                            66$:
7482
7483  024710  062706  000004                  ADD     #4,R6           ;POINT TO WHERE WE TRAPPED FROM.
7484
7485  024714  011637  024764                  MOV     (R6),3$         ;PICK UP LOC
7486  024720  162737  000002  024764          SUB     #2,3$           ;FROM REAL ADDR.
7487  024726  013746  024764                  MOV     3$,-(SP)        ;;SAVE 3$ FOR TYPEOUT
7488  024732  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7489
7490  024734  023727  024762  000004          CMP     2$,#4           ;DID WE TRAP TO LOC 4?
7491  024742  001405                          BEQ     1$              ;IF SO - DON'T RETURN!
7492  024744  023727  024762  000010          CMP     2$,#10          ;DID WE TRAP TO LOC. 10?
7493  024752  001401                          BEQ     1$              ;ID SO - DON'T RETURN!
7494  024754  000002                          RTI                     ;TRY RETURNING.
7495
7496

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

7500  024756  000000                  1$:     HALT
7501  024760  000776                          BR      1$              ;WE STOPPED HERE BECAUSE WE TRAPPED
7502                                                                  ;TO LOC 4 OR LOC 10.  THIS IS A
7503                                                                  ;FATAL CONDITION THAT WE CAN NOT
7504                                                                  ;RECOVER FROM.
7505

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

7509  024762  000000                  2$:     .WORD   0               ;USED BY IOTRP TO STORE WHERE WE TRAPPED TO.
7510  024764  000000                  3$:     .WORD   0               ;USED BY IOTRP TO sTORE WHERE WE TRAPPED FROM.
7511                                          .SBTTL
7512                                          .SBTTL  *SYSMAC ROUTINES
7513                                          .SBTTL
7514
7515                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
7516
7517                                    ;;**************************************************************
7518                                    ;*THIS ROUTINE IS USED TO cHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7519                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
7520                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7521                                    ;*CALL:
7522                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7523                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
7524                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7525                                    ;*      .BYTE   M               ;;M=1 OR 0
7526                                    ;*                                      ;;1=TYPE LEADING ZEROS
7527                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
7528                                    ;*
7529                                    ;*$TYPON----ENTER HERE TO tYPE OUT WITH THE SAME PARAMETERS AS THE LAST
7530                                    ;*$TYPOS OR $TYPOC
7531                                    ;*CALL:
7532                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7533                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
```

```
7534                                    ;*
7535                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
7536                                    ;*CALL:
7537                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7538                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
7539
7540  024766  017646  000000          $TYpOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
7541  024772  116637  000001  025211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
7542  025000  112637  025213                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
7543  025004  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
7544  025010  000406                          BR      $TYPON
7545  025012  112737  000001  025211  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
7546  025020  112737  000006  025213          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
7547  025026  112737  000005  025210  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
7548  025034  010346                          MOV     R3,-(SP)        ;;SAVE R3
7549  025036  010446                          MOV     R4,-(SP)        ;;SAVE R4
7550  025040  010546                          MOV     R5,-(SP)        ;;SAVE R5
7551  025042  113704  025213                  MOVB    $OMoDE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
7552  025046  005404                          NEG     R4
7553  025050  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
7554  025054  110437  025212                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
7555  025060  113704  025211                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
7556  025064  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
7557  025070  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
7558  025072  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
7559  025074  000404                          BR      3$              ;;GO DO MSB
7560  025076  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
7561  025100  006105                          ROL     R5
7562  025102  006105                          ROL     R5
7563  025104  010503                          MOV     R5,R3
7564  025106  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
7565  025110  105337  025212                  DECB    $OMODE          ;;TYPE THIS DIGIT?
7566  025114  100016                          BPL     7$              ;;BR IF NO
7567  025116  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
7568  025122  001002                          BNE     4$              ;;TEST FOR 0
7569  025124  005704                          TST     R4              ;;SUPPRESS THIS 0?
7570  025126  001403                          BEQ     5$              ;;BR IF YES
7571  025130  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
7572  025132  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
7573  025136  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
7574  025142  110337  025206                  MOVB    R3,8$           ;;SAVE FOR TYPING
7575  025146  104401  025206                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
7576  025152  105337  025210          7$:     DECB    $OCNT           ;;COUNT BY 1
7577  025156  003347                          BGT     2$              ;;BR IF MORE TO DO
7578  025160  002402                          BLT     6$              ;;BR IF DONE
7579  025162  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
7580  025164  000744                          BR      2$              ;;GO DO THE LAST DIGIT
7581  025166  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
7582  025170  012604                          MOV     (SP)+,R4        ;;RESTORE R4
7583  025172  012603                          MOV     (SP)+,R3        ;;RESTORE R3
7584  025174  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
7585  025202  012616                          MOV     (SP)+,(SP)
7586  025204  000002                          RTI                     ;;RETURN
7587  025206     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
```

```
7588  025207  000                       .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
7589  025210  000               $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
7590  025211  000               $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
7591  025212  000000            $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO tYPE
7592                            .SBTTL  ERROR HANDLER ROUTINE
7593
7594                            ;;**************************************************************
7595                            ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7596                            ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7597                            ;*AND GO TO $ERRTYP ON ERROR
7598                            ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7599                            ;*SW15=1         HALT ON ERROR
7600                            ;*SW13=1         INHIBIT ERROR TYPEOUTS
7601                            ;*SW10=1         BELL ON ERROR
7602                            ;*SW09=1         LOOP ON ERROR
7603                            ;*CALL
7604                            ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
7605
7606  025214                    $ERrOR:
7607  025214  104407                           CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
7608  025216  105237  001103    7$:           INCB    $ERFLG          ;;SET THE ERROR FLAG
7609  025222  001775                           BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
7610  025224  013777  001102  153710          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
7611  025232  032777  002000  153700          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
7612  025240  001402                           BEQ     1$              ;;NO - SKIP
7613  025242  104401  001172                   TYPE    ,$BELL          ;;RING BELL
7614  025246  005237  001112    1$:           INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
7615  025252  011637  001116                   MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
7616  025256  162737  000002  001116          SUB     #2,$ERRPC
7617  025264  117737  153626  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
7618  025272  032777  020000  153640          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
7619  025300  001004                           BNE     20$             ;;SKIP TYPEOUTS
7620  025302  004737  027420                   JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
7621  025306  104401  001177                   TYPE    ,$CRLF
7622  025312                    20$:
7623  025312  122737  000001  001222          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
7624  025320  001007                           BNE     2$              ;;NO,SKIP APT ERROR REPORT
7625  025322  113737  001114  025334          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
7626  025330  004737  027420                   JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
7627  025334  000               21$:          .BYTE   0
7628  025335  000                           .BYTE   0
7629  025336  000777            22$:          BR      22$             ;;APT ERROR LOOP
7630  025340  005777  153574    2$:           TST     @SWR            ;;HALT ON ERROR
7631  025344  100002                           BPL     3$              ;;SKIP IF CONTINUE
7632  025346  000000                           HALT                    ;;HALT ON ERROR!
7633  025350  104407                           CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
7634  025352  032777  001000  153560  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
7635  025360  001402                           BEQ     4$              ;;BR IF NO
7636  025362  013716  001110                   MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
7637  025366  005737  001170    4$:           TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
7638  025372  001402                           BEQ     5$              ;;BR IF NONE
7639  025374  013716  001170                   MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
7640  025400                    5$:
7641  025400  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.

025402                          $ERRTYP:
025402  104401  001177                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025406  010046                          MOV     R0,-(SP)        ;;SAVE R0
025410  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
025412  153700  001114                  BISB    @#$ITEMB,R0
025416  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
025420  013746  001116                  MOV     $ERRPC,-(SP)    ;;SAVE $ERrPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
025424  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025426  000426                          BR      6$              ;;GET OUT
025430  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
025432  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
025434  006300                          ASL     R0
025436  006300                          ASL     R0
025440  062700  001360                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
025444  012037  025454                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
025450  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
025452  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
025454  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
025456  104401  001177                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025462  012037  025472          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
025466  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
025470  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
025472  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
025474  104401  001177                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025500  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
025502  001004                          BNE     7$              ;;GO TYPE THE DATA
025504  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
025506  104401  001177                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025512  000207                          RTS     PC              ;;RETURN
025514                          7$:
025514  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
025516  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025520  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
025522  001770                          BEQ     6$              ;;BR IF NO
025524  104401  025532                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
025530  000771                          BR      7$              ;;LOOP
025532  020040      000         8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
025536                                  .EVEN
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;**************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
```

```
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
                                        $TYPDS:
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
025550  012746  020200                          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
025554  016605  000020                          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
025560  100004                                  BPL     1$              ;;BR IF INPUT IS POS.
025562  005405                                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
025564  112766  000055  000001                  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
025572  005000                          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
025574  012703  025752                          MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
025600  112723  000040                          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
025604  005002                          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
025606  016001  025742                          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
025612  160105                          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
025614  002402                                  BLT     4$              ;;BR IF DONE
025616  005202                                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
025620  000774                                  BR      3$
025622  060105                          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
025624  005702                                  TST     R2              ;;CHECK IF BCD DIGIT=0
025626  001002                                  BNE     5$              ;;FALL THROUGH IF 0
025630  105716                                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
025632  100407                                  BMI     7$              ;;BR IF YES
025634  106316                          5$:     ASLB    (SP)            ;;MSD?
025636  103003                                  BCC     6$              ;;BR IF NO
025640  116663  000001  177777                  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
025646  052702  000060                  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
025652  052702  000040                  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
025656  110223                                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
025660  005720                                  TST     (R0)+           ;;JUST INCREMENTING
025662  020027  000010                          CMP     R0,#10          ;;CHECK THE TABLE INDEX
025666  002746                                  BLT     2$              ;;GO DO THE NEXT DIGIT
025670  003002                                  BGT     8$              ;;GO TO EXIT
025672  010502                                  MOV     R5,R2           ;;GET THE LSD
025674  000764                                  BR      6$              ;;GO CHANGE TO ASCII
025676  105726                          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
025700  100003                                  BPL     9$              ;;BR IF NO
025702  116663  177777  177776                  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
025710  105013                          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
025712  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
025714  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
025716  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
025720  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
025722  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
025724  104401  025752                          TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
025730  016666  000002  000004                  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
025736  012616                                  MOV     (SP)+,(SP)
```

```
7750  025740  000002                          RTI                     ;;RETURN TO uSER
7751  025742  023420                $DTBL:    10000.
7752  025744  001750                          1000.
7753  025746  000144                          100.
7754  025750  000012                          10.
7755  025752  000004                $DBLK:    .BLKW   4
7756                                .SBTTL  SCOPE HANDLER ROUTINE
7757
7758                                ;;*****************************************************************
7759                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
7760                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
7761                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
7762                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7763                                ;*SW14=1          LOOP ON TEST
7764                                ;*SW11=1          INHIBIT ITERATIONS
7765                                ;*SW09=1          LOOP ON ERROR
7766                                ;*SW08=1          LOOP ON TEST IN SWR<7:0>
7767                                ;*CALL
7768                                ;*      SCOPE               ;;SCOPE=IOT
7769
7770  025762                        $SCOPE:
7771  025762  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
7772  025764  032777  040000  153146 1$:      BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
7773  025772  001114                         BNE     $OVER           ;;YES IF SW14=1
7774                                ;#####START OF CODE FOR THE XOR TESTER#####
7775  025774  000416                $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
7776                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
7777  025776  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
7778  026002  012737  026022  000004        MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
7779  026010  005737  177060                TST     @#177060        ;;TIME OUT ON XOR?
7780  026014  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
7781  026020  000463                        BR      $SVLAD          ;;GO TO THE NEXT TEST
7782  026022  022626                5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
7783  026024  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
7784  026030  000423                        BR      7$              ;;LOOP ON THE PRESENT TEST
7785  026032                        6$:;#####END OF CODE FOR THE XOR TESTER#####
7786  026032  032777  000400  153100        BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
7787  026040  001404                        BEQ     2$              ;;BR IF NO
7788  026042  127737  153072  001102        CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
7789  026050  001465                        BEQ     $OVER           ;;BR IF YES
7790  026052  105737  001103         2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
7791  026056  001421                        BEQ     3$              ;;BR IF NO
7792  026060  123737  001115  001103        CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
7793  026066  101015                        BHI     3$              ;;BR IF NO
7794  026070  032777  001000  153042        BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
7795  026076  001404                        BEQ     4$              ;;BR IF NO
7796  026100  013737  001110  001106 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
7797  026106  000446                        BR      $OVER
7798  026110  105037  001103         4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
7799  026114  005037  001166                CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO mAKE
7800  026120  000415                        BR      1$              ;;ESCAPE TO THE NEXT TEST
7801  026122  032777  004000  153010 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
7802  026130  001011                        BNE     1$              ;;BR IF YES
7803  026132  005737  001210                TST     $PASS           ;;IF FIRST PASS OF PROGRAM
```

```
7804  026136  001406                    BEQ     1$                ;;       INHIBIT ITERATIONS
7805  026140  005237  001104            INC     $ICNT             ;;INCREMENT ITERATION COUNT
7806  026144  023737  001166  001104    CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
7807  026152  002024                    BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
7808  026154  012737  000001  001104  1$: MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
7809  026162  013737  026240  001166    MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
7810  026170  105237  001102          $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
7811  026174  113737  001102  001206    MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
7812  026202  011637  001106            MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
7813  026206  011637  001110            MOV     (SP),$LPERR       ;;SAVE ERROR LOOP ADDRESS
7814  026212  005037  001170            CLR     $ESCAPE           ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
7815  026216  112737  000001  001115    MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7816  026224  013777  001102  152710  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
7817  026232  013716  001106            MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
7818  026236  000002                    RTI                       ;;FIXES PS
7819  026240  000012                  $MXCNT: 10.                       ;;MAX. NUMBER OF ITERATIONS
7820                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
7821
7822                                   ;;**********************************************************************
7823                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
7824                                   ;*CHANGE IT TO BINARY.
7825                                   ;*CALL:
7826                                   ;*      RDOCT                     ;;READ AN OCTAL NUMBER
7827                                   ;*      RETURN HERE               ;;LOW ORDER BITS ARE ON TOP OF THE STACK
7828                                   ;*                                ;;HIGH ORDER BITS ARE IN $HIOCT
7829
7830  026242  011646                  $RDOCT: MOV     (SP),-(SP)        ;;PROVIDE SPACE FOR THE
7831  026244  016666  000004  000002          MOV     4(SP),2(SP)       ;;INPUT NUMBER
7832  026252  010046                          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
7833  026254  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
7834  026256  010246                          MOV     R2,-(SP)          ;;PUSH R2 ON STACK
7835  026260  104411                  1$:     RDLIN                     ;;READ AN ASCIZ LINE
7836  026262  012600                          MOV     (SP)+,R0          ;;GET ADDRESS OF 1ST CHARACTER
7837  026264  005001                          CLR     R1                ;;CLEAR DATA WORD
7838  026266  005002                          CLR     R2
7839  026270  112046                  2$:     MOVB    (R0)+,-(SP)       ;;PICKUP THIS CHARACTER
7840  026272  001412                          BEQ     3$                ;;IF ZERO GET OUT
7841  026274  006301                          ASL     R1                ;;*2
7842  026276  006102                          ROL     R2
7843  026300  006301                          ASL     R1                ;;*4
7844  026302  006102                          ROL     R2
7845  026304  006301                          ASL     R1                ;;*8
7846  026306  006102                          ROL     R2
7847  026310  042716  177770                  BIC     #^C7,(SP)         ;;STRIP THE ASCII JUNK
7848  026314  062601                          ADD     (SP)+,R1          ;;ADD IN THIS DIGIT
7849  026316  000764                          BR      2$                ;;LOOP
7850  026320  005726                  3$:     TST     (SP)+             ;;CLEAN TERmINATOR FROM STACK
7851  026322  010166  000012                  MOV     R1,12(SP)         ;;SAVE THE RESULT
7852  026326  010237  026342                  MOV     R2,$HIOCT
7853  026332  012602                          MOV     (SP)+,R2          ;;POP STACK INTO R2
7854  026334  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
7855  026336  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
7856  026340  000002                          RTI                       ;;RETURN
7857  026342  000000                  $HIOCT: .WORD   0                 ;;HIGH ORDER BITS GO HERE
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
                                        ;;*****************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
026344  022737  000176  001140  $CKSWR: CMP     #$WREG,SWR      ;;IS THE SOFT-SWR SELECTED?
026352  001074                          BNE     15$             ;;BRANCH IF NO
026354  105777  152564                  TSTB    @$TKS           ;;CHAR THERE?
026360  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
026362  117746  152560                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
026366  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
026372  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
026376  001062                          BNE     15$             ;;NO, RETURN TO USER
026400  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
026406  001456                          BEQ     15$             ;;BRANCH IF YES

026410  104401  027071                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
026414  104401  027076          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
026420  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
026424  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
026426  104401  027107                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
026432  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
026434  005046                          CLR     -(SP)           ;;THE NEW SWR
026436  105777  152502          7$:     TSTB    @$TKS           ;;CHAR THERE?
026442  100375                          BPL     7$              ;;IF NOT TRY AGAIN

026444  117746  152476                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
026450  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


                                        ;;*****************************************************************
026454  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
026460  001005                          BNE     10$             ;;BRANCH IF NOT
026462  104401  027064                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
026466  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
026472  000757                          BR      19$             ;;LET'S TRY IT AGAIN


026474  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
026500  001022                          BNE     16$             ;;BRANCH IF NO
026502  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
026506  001403                          BEQ     11$             ;;BRANCH IF YES
026510  016677  000002  152422          MOV     2(SP),@SWR      ;;SAVE NEW SWR
026516  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
026522  104401  001177          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
026526  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
026534  001003                          BNE     15$             ;;BRANCH IF NOT
026536  012777  000100  152400          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
026544  000002                  15$:    RTI                     ;;RETURN
```

```
7912  026546  004737  027332          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
7913  026552  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
7914  026556  002420                          BLT     18$             ;;BRANCH IF YES
7915  026560  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
7916  026564  003015                          BGT     18$             ;;BRANCH IF YES
7917  026566  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
7918  026572  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
7919  026576  001403                          BEQ     17$             ;;BRANCH IF YES
7920  026600  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
7921  026602  006316                          ASL     (SP)            ;;   CHAR OVER TO mAKE
7922  026604  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
7923  026606  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
7924  026612  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
7925  026616  000707                          BR      7$              ;;GET THE NEXT ONE
7926  026620  104401  001176          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
7927  026624  000720                          BR      20$             ;;SIMULATE CONTROL-U
7928                                  .DSABL  LSB
7929
7930
7931                                  ;;*****************************************************************
7932                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7933                                  ;*CALL:
7934                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
7935                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
7936                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
7937                                  ;
7938
7939  026626  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
7940  026630  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
7941  026636  105777  152302          1$:     TSTB    @$TKS           ;;WAIT FOR
7942  026642  100375                          BPL     1$              ;;A CHARACTER
7943  026644  117766  152276  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
7944  026652  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
7945  026660  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
7946  026666  001013                          BNE     3$              ;;BRANCH IF NO
7947  026670  105777  152250          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
7948  026674  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
7949  026676  117746  152244                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
7950  026702  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
7951  026706  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
7952  026712  001366                          BNE     2$              ;;IF NOT DISCARD IT
7953  026714  000750                          BR      1$              ;;YES, RESUME
7954  026716  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
7955  026724  002407                          BLT     4$              ;;BRANCH IF YES
7956  026726  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
7957  026734  003003                          BGT     4$              ;;BRANCH IF YES
7958  026736  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
7959  026744  000002                  4$:     RTI                     ;;GO BACK TO uSER
7960                                  ;;*****************************************************************
7961                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
7962                                  ;*CALL:
7963                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
7964                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
7965                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
```

```
                                                ;
                                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
                                1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
                                2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
                                        BLOS    4$              ;;BR IF YES
                                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
                                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
                                10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
                                        BNE     3$              ;;SKIP IF NOT
                                4$:     TYPE    ,$QUES          ;;TYPE A '?'
                                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
                                3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
                                        TYPE    ,9$
                                        CMPB    #15,(R3)+       ;;CHECK FOR RETURN
                                        BNE     2$              ;;LOOP IF NOT RETURN
                                        CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
                                        TYPE    ,$LF            ;;TYPE A LINE FEED
                                        MOV     (SP)+,R3        ;;RESTORE R3
                                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
                                        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
                                        MOV     #$TTYIN,4(SP)
                                        RTI                     ;;RETURN
                                9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
                                        .BYTE   0               ;;TERMINATOR
                                $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
                                $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
                                $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
                                $MSWR:  .ASCIZ  <15><12>/SWR = /
                                $MNEW:  .ASCIZ  /  NEW = /
                                .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
                                ;*
                                $TYpE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
                                        BPL     1$              ;;BR IF YES
                                        HALT                    ;;HALT HERE IF NO TERMINAL
                                        BR      3$              ;;LEAVE
                                1$:     MOV     R0,-(SP)        ;;SAVE R0
                                        MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
```

```
8020  027140  122737  000001  001222          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
8021  027146  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
8022  027150  132737  000100  001223          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
8023  027156  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
8024  027160  010037  027170                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
8025  027164  004737  027410                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
8026  027170  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
8027  027172  132737  000040  001223  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
8028  027200  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
8029  027202  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
8030  027204  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
8031  027206  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
8032  027210  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
8033  027212  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
8034  027216  000002                          RTI                     ;;RETURN
8035  027220  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
8036  027224  001430                          BEQ     8$
8037  027226  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
8038  027232  001006                          BNE     5$
8039  027234  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
8040  027236  104401                          TYPE                    ;;TYPE A CR AND LF
8041  027240  001177                          $CRLF
8042  027242  105037  027376                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
8043  027246  000755                          BR      2$              ;;GET NEXT CHARACTER
8044  027250  004737  027332          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
8045  027254  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
8046  027260  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
8047  027262  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
8048                                                                  ;;AND THE NULL CHAR.
8049  027266  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
8050  027272  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
8051  027274  004737  027332                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
8052  027300  105337  027376                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
8053  027304  000770                          BR      7$              ;;LOOP
8054
8055                                  ;HORIZONTAL TAB PROCESSOR
8056
8057  027306  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
8058  027312  004737  027332          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
8059  027316  132737  000007  027376          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
8060  027324  001372                          BNE     9$              ;;TAB STOP
8061  027326  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
8062  027330  000724                          BR      2$              ;;GET NEXT CHARACTER
8063  027332  105777  151612          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
8064  027336  100375                          BPL     $TYPEC
8065  027340  116677  000002  151604          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
8066  027346  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
8067  027354  001003                          BNE     1$              ;;BRANCH IF NO
8068  027356  105037  027376                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
8069  027362  000406                          BR      $TYPEX          ;;EXIT
8070  027364  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
8071  027372  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
8072  027374  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
8073  027376  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
```

```
 8074  027400  000207                  $TYPEX: RTS     PC
 8075
 8076                                  .SBTTL  APT COMMUNICATIONS ROUTINE
 8077
 8078                                  ;;*****************************************************************
 8079  027402  112737  000001  027646  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 8080  027410  112737  000001  027644  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 8081  027416  000403                          BR      $ATYC
 8082  027420  112737  000001  027646  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 8083  027426                          $ATYC:
 8084  027426  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8085  027430  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8086  027432  105737  027644                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 8087  027436  001450                          BEQ     5$              ;;IF NOT:  BR
 8088  027440  122737  000001  001222          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 8089  027446  001031                          BNE     3$              ;;IF NOT:  BR
 8090  027450  132737  000100  001223          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 8091  027456  001425                          BEQ     3$              ;;IF NOT:  BR
 8092  027460  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 8093  027464  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 8094  027472  005737  001202          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 8095  027476  001375                          BNE     1$              ;;IF NOT:  WAIT
 8096  027500  010037  001216                  MOV     R0,$MSgAD       ;;PUT ADDR IN MAILBOX
 8097  027504  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 8098  027506  001376                          BNE     2$
 8099  027510  163700  001216                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 8100  027514  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 8101  027516  010037  001220                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 8102  027522  012737  000004  001202          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 8103  027530  000413                          BR      5$
 8104  027532  017637  000004  027556  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 8105  027540  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 8106  027546  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 8107  027552  004737  027120                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
 8108  027556  000000                  4$:     .WORD   0
 8109  027560                          5$:
 8110  027560  105737  027646          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 8111  027564  001416                          BEQ     12$             ;;IF NOT:  BR
 8112  027566  005737  001222                  TST     $ENV            ;;RUNNING UNDER APT?
 8113  027572  001413                          BEQ     12$             ;;IF NOT:  BR
 8114  027574  005737  001202          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 8115  027600  001375                          BNE     11$             ;;IF NOT:  WAIT
 8116  027602  017637  000004  001204          MOV     @4(SP),$FATAL   ;;GET ERROR #
 8117  027610  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 8118  027616  005237  001202                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 8119  027622  105037  027646          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 8120  027626  105037  027645                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 8121  027632  105037  027644                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 8122  027636  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8123  027640  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 8124  027642  000207                          RTS     PC              ;;RETURN
 8125  027644     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
 8126  027645     000          $LFLG:  .BYTE   0               ;;LOG FLAG
 8127  027646     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
```

```
                                        .EVEN
                                APTSIZE=200
                                APTENV=001
                                APTSPOOL=100
                                APTCSUP=040
                                .SBTTL  POWER DOWn AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
027650  012737  030010  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
027656  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
027664  010046                        MOV     R0,-(SP)         ;;PUSH R0 ON STACK
027666  010146                        MOV     R1,-(SP)         ;;PUSH R1 ON STACK
027670  010246                        MOV     R2,-(SP)         ;;PUSH R2 ON STACK
027672  010346                        MOV     R3,-(SP)         ;;PUSH R3 ON STACK
027674  010446                        MOV     R4,-(SP)         ;;PUSH R4 ON STACK
027676  010546                        MOV     R5,-(SP)         ;;PUSH R5 ON STACK
027700  017746  151234                MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
027704  010637  030014                MOV     SP,$SAVR6        ;;SAVE SP
027710  012737  027722  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
027716  000000                        HALT
027720  000776                        BR      .-2              ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
027722  012737  030010  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
027730  013706  030014                MOV     $SAVR6,SP        ;;GET SP
027734  005037  030014                CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
027740  005237  030014          1$:   INC     $SAVR6           ;;WAIT FOR THE INC
027744  001375                        BNE     1$               ;;OF  WORD
027746  012677  151166                MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
027752  012605                        MOV     (SP)+,R5         ;;POP STACK INTO R5
027754  012604                        MOV     (SP)+,R4         ;;POP STACK INTO R4
027756  012603                        MOV     (SP)+,R3         ;;POP STACK INTO R3
027760  012602                        MOV     (SP)+,R2         ;;POP STACK INTO R2
027762  012601                        MOV     (SP)+,R1         ;;POP STACK INTO R1
027764  012600                        MOV     (SP)+,R0         ;;POP STACK INTO R0
027766  012737  027650  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
027774  012737  000340  000026        MOV     #340,@#PWRVEC+2  ;;PRIO:7
030002  104401                        TYPE                     ;REPORT THE POWER FAILURE
030004  030016          $PWRMG: .WORD   $POWER                   ;;POWER FAIL MESSAGE POINTER
030006  000002                        RTI
030010  000000          $ILLUP: HALT                             ;;THE POWER UP SEQUENCE WAS STARTED
030012  000776                        BR      .-2              ;;  BEFORE THE POWER DOWn WAS COMPLETE
030014  000000          $SAVR6: 0                                ;;PUT THE SP HERE
030016  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
030024  000122
                                        .EVEN
                                .SBTTL  TRAP DECODER

                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
```

```
                                ;*GO TO THAT ROUTINE.
                                ;
030026  010046                $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
030030  016600  000002                MOV     2(SP),R0        ;;GET TRAP ADDRESS
030034  005740                        TST     -(R0)           ;;BACKUP BY 2
030036  111000                        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
030040  006300                        ASL     R0              ;;POSITION FOR INDEXING
030042  016000  030062                MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
030046  000200                        RTS     R0              ;;GO TO rOUTINE
                                ;
                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
                                ;
030050  011646                $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
030052  016666  000004  000002        MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
030060  000002                        RTI                     ;;RESTORE THE PSW
                                ;
                                .SBTTL  TRAP TABLE
                                ;
                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.
                                ;
                                ;       ROUTINE
                                ;       -------
030062  030050                $TRPAD: .WORD   $TRAP2
030064  027120                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
030066  025012                        $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
030070  024766                        $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
030072  025026                        $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
030074  025536                        $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
                                ;
030076  026414                        $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
                                ;
030100  026344                        $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
030102  026626                        $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
030104  026746                        $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
030106  026242                        $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
030110  024612                        IOTRD   ;;CALL=IOTT     TRAP+13(104413)
030112  005015  046103  041517  EM1:  .ASCIZ  <15><12>/CLOCKA SR FUNCTION ERROR/
030120  040513  051440  020122
030126  052506  041516  044524
030134  047117  042440  051122
030142  051117     000
030145     015  041412  047514  EM2:  .ASCIZ  <15><12>/CLOCKA SR DATA ERROR/
030152  045503  020101  051123
030160  042040  052101  020101
030166  051105  047522  000122
030174  005015  046103  041517  EM3:  .ASCIZ  <15><12>/CLOCKA BR DATA ERROR/
030202  040513  041040  020122
030210  040504  040524  042440
030216  051122  051117     000
030223     015  041412  047514  EM4:  .ASCIZ  <15><12>/CLOCKA CR DATA ERROR/
030230  045503  020101  051103
030236  042040  052101  020101
```

```
 8236  030244  051105  047522  000122
 8237  030252  005015  046103  041517  EM5:    .ASCIZ  <15><12>/CLOCKB SR DATA ERROR/
 8238  030260  041113  051440  020122
 8239  030266  040504  040524  042440
 8240  030274  051122  051117     000
 8241  030301     015  041412  047514  EM6:    .ASCIZ  <15><12>/CLOCKB BR DATA ERROR/
 8242  030306  045503  020102  051102
 8243  030314  042040  052101  020101
 8244  030322  051105  047522  000122
 8245  030330  005015  046103  041517  EM7:    .ASCIZ  <15><12>/CLOCKB CR DATA ERROR/
 8246  030336  041113  041440  020122
 8247  030344  040504  040524  042440
 8248  030352  051122  051117     000
 8249  030357     015  042012  040525  EM10:   .ASCIZ  <15><12>/DUAL ADDRESS ERROR/
 8250  030364  020114  042101  051104
 8251  030372  051505  020123  051105
 8252  030400  047522  000122
 8253  030404  005015  046103  041517  EM11:   .ASCIZ  <15><12>#CLOCK A COUNT ERROR #
 8254  030412  020113  020101  047503
 8255  030420  047125  020124  051105
 8256  030426  047522  020122     000
 8257  030433     015  041412  047514  EM12:   .ASCIZ  <15><12>#CLOCK A COUNT FUNCTION ERROR #
 8258  030440  045503  040440  041440
 8259  030446  052517  052116  043040
 8260  030454  047125  052103  047511
 8261  030462  020116  051105  047522
 8262  030470  020122     000
 8263  030473     015  041412  047514  EM14:   .ASCIZ  <15><12>#CLOCK B COUNT FUNCTION ERROR #
 8264  030500  045503  041040  041440
 8265  030506  052517  052116  043040
 8266  030514  047125  052103  047511
 8267  030522  020116  051105  047522
 8268  030530  020122     000
 8269  030533     015  041412  047514  EM15:   .ASCIZ  <15><12>#CLOCK B COUNT ERROR #
 8270  030540  045503  041040  041440
 8271  030546  052517  052116  042440
 8272  030554  051122  051117  000040
 8273  030562  005015  046103  041517  EM16:   .ASCIZ  <15><12>#CLOCK A INTERRUPT ERROR #
 8274  030570  020113  020101  047111
 8275  030576  042524  051122  050125
 8276  030604  020124  051105  047522
 8277  030612  020122     000
 8278  030615     015  041412  047514  EM17:   .ASCIZ  <15><12>#CLOCK B INTERRUPT ERROR #
 8279  030622  045503  041040  044440
 8280  030630  052116  051105  052522
 8281  030636  052120  042440  051122
 8282  030644  051117  000040
 8283  030650  005015  046103  041517  EM20:   .ASCIZ  <15><12>#CLOCK A REPEATABILITY ERROR #
 8284  030656  020113  020101  042522
 8285  030664  042520  052101  041101
 8286  030672  046111  052111  020131
 8287  030700  051105  047522  020122
 8288  030706     000
 8289  030707     015  041412  047514  EM23:   .ASCIZ  <15><12>#CLOCK B REPEATABILITY ERROR #
```

```
 8290  030714  045503  041040  051040
 8291  030722  050105  040505  040524
 8292  030730  044502  044514  054524
 8293  030736  042440  051122  051117
 8294  030744  000040
 8295  030746  005015  046103  041517  EM26:   .ASCIZ  <15><12>#CLOCK ADDRESSING ERROR#
 8296  030754  020113  042101  051104
 8297  030762  051505  044523  043516
 8298  030770  042440  051122  051117
 8299  030776     000
 8300
 8301  030777     015  042412  051122  DH1:    .ASCIZ  <15><12>#ERRPC   ASR     WAS     S/B#
 8302  031004  041520  020040  040440
 8303  031012  051123  020040  020040
 8304  031020  053440  051501  020040
 8305  031026  020040  051440  041057
 8306  031034     000
 8307  031035     015  042412  051122  DH3:    .ASCIZ  <15><12>#ERRPC   ABR     WAS     S/B#
 8308  031042  041520  020040  040440
 8309  031050  051102  020040  020040
 8310  031056  053440  051501  020040
 8311  031064  020040  051440  041057
 8312  031072     000
 8313  031073     015  042412  051122  DH4:    .ASCIZ  <15><12>#ERRPC   ACR     WAS     S/B#
 8314  031100  041520  020040  040440
 8315  031106  051103  020040  020040
 8316  031114  053440  051501  020040
 8317  031122  020040  051440  041057
 8318  031130     000
 8319  031131     015  042412  051122  DH5:    .ASCIZ  <15><12>#ERRPC   BSR     WAS     S/B#
 8320  031136  041520  020040  041040
 8321  031144  051123  020040  020040
 8322  031152  053440  051501  020040
 8323  031160  020040  051440  041057
 8324  031166     000
 8325  031167     015  042412  051122  DH6:    .ASCIZ  <15><12>#ERRPC   BBR     WAS     S/B#
 8326  031174  041520  020040  041040
 8327  031202  051102  020040  020040
 8328  031210  053440  051501  020040
 8329  031216  020040  051440  041057
 8330  031224     000
 8331  031225     015  042412  051122  DH7:    .ASCIZ  <15><12>#ERRPC   BCR     WAS     S/B#
 8332  031232  041520  020040  041040
 8333  031240  051103  020040  020040
 8334  031246  053440  051501  020040
 8335  031254  020040  051440  041057
 8336  031262     000
 8337  031263     015  042412  051122  DH10:   .ASCII  <15><12>/ERROR   GOOD    BAD     GOOD    DATA READ FROM/
 8338  031270  051117  020040  043440
 8339  031276  047517  020104  020040
 8340  031304  041040  042101  020040
 8341  031312  020040  043440  047517
 8342  031320  020104  020040  042040
 8343  031326  052101  020101  042522
```

```
 8344  031334  042101  043040  047522
 8345  031342     115
 8346  031343     015  020012  050040          .ASCIZ  <15><12>/  PC    ADDR    ADDR    DATA    DUAL ADDRESS/
 8347  031350  020103  020040  040440
 8348  031356  042104  020122  020040
 8349  031364  040440  042104  020122
 8350  031372  020040  042040  052101
 8351  031400  020101  020040  042040
 8352  031406  040525  020114  042101
 8353  031414  051104  051505  000123
 8354  031422  005015  051105  050122  DH12:   .ASCIZ  <15><12>#ERRPC   ASR #
 8355  031430  020103  020040  051501
 8356  031436  020122     000
 8357  031441     015  042412  051122  DH14:   .ASCIZ  <15><12>#ERRPC   BSR#
 8358  031446  041520  020040  041040
 8359  031454  051123     000
 8360  031457     015  042412  051122  DH20:   .ASCIZ  <15><12>#ERRPC   ASR     2NDCNT  1STCNT #
 8361  031464  041520  020040  040440
 8362  031472  051123  020040  020040
 8363  031500  031040  042116  047103
 8364  031506  020124  030440  052123
 8365  031514  047103  020124     000
 8366  031521     015  042412  051122  DH23:   .ASCIZ  <15><12>#ERRPC   BSR     2NDCNT  1STCNT #
 8367  031526  041520  020040  041040
 8368  031534  051123  020040  020040
 8369  031542  031040  042116  047103
 8370  031550  020124  030440  052123
 8371  031556  047103  020124     000
 8372  031563     015  042412  051122  DH26:   .ASCIZ  <15><12>#ERRPC   CLOCK ADDR.#
 8373  031570  041520  020040  041440
 8374  031576  047514  045503  040440
 8375  031604  042104  027122     000
 8376
 8377          031612                          .EVEN
 8378
 8379  031612  001116  001326  001126  DT1:    .WORD   $ERrPC,ASR,$BDDAT,$GDDAT,0
 8380  031620  001124  000000
 8381  031624  001116  001330  001126  DT3:    .WORD   $ERRPC,ABR,$BDDAT,$GDDAT,0
 8382  031632  001124  000000
 8383  031636  001116  001332  001126  DT4:    .WORD   $ERRPC,ACR,$BDDAT,$GDDAT,0
 8384  031644  001124  000000
 8385  031650  001116  001334  001126  DT5:    .WORD   $ERRPC,BSR,$BDDAT,$GDDAT,0
 8386  031656  001124  000000
 8387  031662  001116  001336  001126  DT6:    .WORD   $ERRPC,BBR,$BDDAT,$GDDAT,0
 8388  031670  001124  000000
 8389  031674  001116  001340  001126  DT7:    .WORD   $ERRPC,BCR,$BDDAT,$GDDAT,0
 8390  031702  001124  000000
 8391  031706  001116  001120  001122  DT10:   .WORD   $ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT,0
 8392  031714  001124  001126  000000
 8393  031722  001116  001326  000000  DT12:   .WORD   $ERRPC,ASR,0
 8394  031730  001116  001334  000000  DT14:   .WORD   $ERRPC,BSR,0
 8395  031736  001116  001332  001124  DT21:   .WORD   $ERRPC,ACR,$GDDAT,$TMp0,0
 8396  031744  001164  000000
 8397  031750  001116  001332  001126  DT22:   .WORD   $ERrPC,ACR,$BDDAT,$TMP0,0
```

```
 8398  031756  001164  000000
 8399  031762  001116  001340  001124  DT24:   .WORD   $ERRPC,BCR,$GDDAT,$TMP0,0
 8400  031770  001164  000000
 8401  031774  001116  001340  001126  DT25:   .WORD   $ERRPC,BCR,$BDDAT,$TMP0,0
 8402  032002  001164  000000
 8403  032006  001116  001164  000000  DT26:   .WORD   $ERRPC,$TMP0,0
 8404
 8405  032014  000000  000000          DF0:    .WORD   0,0
 8406
 8407
 8408
 8409                                  ;*
 8410                                  ;*THIS SUB CODE IS USED TO INITIALIZE THE LPA-11
 8411                                  ;*FIRST WE WILL LOAD MICROCODE INTO KMC-11
 8412                                  ;*NEXT WE WILL INIT BOTH UPROCESSORS
 8413                                  ;*THEN WE WILL LOAD DEVICE TABLE IN SLAVE UP.
 8414                                  ;*THE ORDER OF LOAD IS DETERMINED BY THE USER.
 8415                                  ;*
 8416                                  ;*      CALL=   JSR     R5,$LPAI
 8417                                  ;*              .WORD   0               ;ADDR. OF DEVICE ADDRESS.
 8418                                  ;* ROUTINES REQUIRED:   .LOADLP
 8419                                  ;* PROGRAMS REQUIRED:   DRLPX2
 8420                                  ;*
 8421
 8422                                  ;*              ;RETURNS WITH $AERR=1 IF SLAVE
 8423                                  ;*              ;MICRO SAYS AN ADDR. DOEs NOT EXSIST. IN THE LIST.
 8424                                  ;*
 8425  032020                  $LPAI:
 8426  032020  013746  000004         MOV     4,-(SP)
 8427
 8428  032024  000413                 BR      31$             ;FIELD DOES NOT HAVE A BUS SWITCH TO
 8429                                                          ;WORRY ABOUT,SO WE WILL UNCONDITIONALLY
 8430                                                          ;BRANCH ARROUD THE NEXT CODE THAT
 8431                                                          ;WORKS BASED ON A BUS SWITCH.
 8432                                                          ;CODE LEFT IN HERE FOR IN HOUSE
 8433                                                          ;PERSONAL WHO MAY PATCH THIS BRANCH
 8434                                                          ;INSTRUCTION TO A <NOP> OCTAL <240>
 8435                                                          ;IN ORDER TO RUN PROGRAM WITH A SWITCH.
 8436
 8437                                                          ;NOTE THIS "SWITCH" IS A PIECE OF INHOUSE
 8438                                                          ;TEST EQUIPMENT ONLY IT CONNECTS
 8439                                                          ;THE UNIBUS TO THE I/O BUS FOR
 8440                                                          ;CERTAIN TESTING.
 8441  032026  012737  032052  000004         MOV     #30$,4
 8442  032034  005237  170000                INC     170000
 8443  032040  104401  032046                TYPE    ,65$            ;;TYPE ASCIZ STRING
 8444  032044  000401                 BR      64$             ;;GET OVER THE ASCIZ
 8445                                  ;;65$:  .ASCIZ  <7>##
 8446  032050                          64$:
 8447  032050  000401                 BR      31$
 8448  032052  022626                 30$:    CMP     (SP)+,(SP)+
 8449  032054  012637  000004         31$:    MOV     (SP)+,4         ;ALL THIS JUNK MUST BE REMOVED!!
 8450  032060  005037  032676                CLR     $AERR
 8451  032064  004537  032700                JSR     R5,$LOAD        ;LOAD MICRO-CODE.
```

```
8452  032070  000000G                          .WORD   DRLPX2          ;FILE "DRLPX2.OBJ"
8453
8454  032072  052777  040000  147540          BIS     #BIT14,@KMADO   ;ISSUE KMC+DMC INIT.
8455
8456  032100                          1$:
8457                                                                  ;"HANGS" HERE THEN KMC-11 ERROR.
8458  032100  010146                          MOV     R1,-(SP)
8459  032102  005001                          CLR     R1
8460  032104  005201                  2$:     INC     R1              ;STALL FOR DMC-UP
8461  032106  001376                          BNE     2$
8462  032110  012777  104000  147522          MOV     #BIT15!BIT11,@KMADO     ;SET RUN, AND ENABLE ARBITRATION.
8463  032116  105201                  25$:    INCB    R1
8464  032120  001376                          BNE     25$
8465
8466  032122  032777  000040  147510          BIT     #BIT5,@KMADO    ;SLAVE READY? (READING IPBM SR)
8467  032130  001401                          BEQ     3$
8468                                                                  ;FATAL LPA-11 ERROR SLAVE NOT READY.
8469  032132  104000                          ERROR
8470
8471  032134  012777  000004  147502  3$:     MOV     #4,@KMAD2       ;READ FAST PATH
8472  032142                          4$:
8473  032142  004537  033610                  JSR     R5, $TOUt       ;-TOUT-CHECK FOR TIMEOUT
8474
8475  032146  104000                          ERROR                   ;/TIME-OUT ERROR
8476                                                                  ;/WE FAILED TO COMPLETE
8477                                                                  ;/CURRENT OPERATION.
8478                                                                  ;/CONTINUES IN THIS LOOP
8479                                                                  ;/WOULD MAKE US "HANG" HERE
8480
8481  032150  000774                          BR              4$
8482
8483                                                                  ;/RETURNS HERE-FROM-TIMED OUT.
8484  032152  122777  000377  147464          CMPB    #377,@KMAD2     ;WAIT TILL KMC DONE COMMAND.
8485  032160  001370                          BNE     4$
8486  032162  122777  000377  147460          CMPB    #377,@KMAD4     ;IF FAST PATH=377 THEN ERROR.
8487  032170  001001                          BNE     35$
8488  032172  104000                          ERROR                   ;IPBM ERROR (SLAVE SIDE)
8489                                                                  ;YOU MUST RUN IPBM DIAGNOSTIC.
8490
8491  032174  122777  000004  147446  35$:    CMPB    #4,@KMAD4       ;IS THIS THE CORRECT VERSION OF MICRO-CODE?
8492  032202  001543                          BEQ     5$              ;YES-CONTINUE.
8493  032204  005227  177777                  INC     #-1
8494  032210  001140                          BNE     5$
8495  032212  005227  177777                  INC     #-1
8496  032216  001135                          BNE     5$
8497  032220  104401  032226                  TYPE    ,67$            ;;TYPE ASCIZ STRING
8498  032224  000440                          BR      66$             ;;GET OVER THE ASCIZ
8499                                  ;;67$:  .ASCIZ  <200>"W A R N I N G  THIS PROGRAM WAS DESIGNED TO rUN WITH VERSION 4"
8500  032326                          66$:
8501  032326  104401  032334                  TYPE    ,69$            ;;TYPE ASCIZ STRING
8502  032332  000430                          BR      68$             ;;GET OVER THE ASCIZ
8503                                  ;;69$:  .ASCIZ  <200>"MICRO-CODE. ANOTHER VERSION CODE WAS DETECTED."
8504  032414                          68$:
8505  032414  104401  032422                  TYPE    ,71$            ;;TYPE ASCIZ STRING
```

```
8506  032420  000434                          BR      70$             ;;GET OVER THE ASCIZ
8507                                  ;;71$:  .ASCIZ  <200>"THIS MAY OR MAYNOT CAUSE FALSE ERROR TO BE REPORTED."<200><200>
8508  032512                          70$:
8509
8510  032512  112737  177777  032644  5$:     MOVB    #0-1,11$        ;DAC CODE FOR SLAVE.
8511  032520  012501                          MOV     (5)+,R1         ;GET NEXT DEVICE ADDR.
8512  032522  021127  000000          6$:     CMP     (R1),#0         ;TERM REACHED?
8513  032526  001444                          BEQ     10$
8514  032530  105237  032644                  INCB    11$
8515  032534  113777  032644  147106          MOVB    11$,@KMAD4      ;FIFO DATA
8516  032542  004737  032646                  JSR     PC,20$          ;ISSUE SEND
8517  032546  112177  147076                  MOVB    (R1)+,@KMAD4    ;SEND LOW BYTE OF DEVICE ADDR TO SLAVE.
8518  032552  004737  032646                  JSR     PC,20$          ;ISSUE SEND
8519  032556  112177  147066                  MOVB    (R1)+,@KMAD4    ;SEND HIGH BYTE OF DEVICE ADDR. TO SLAVE.
8520  032562  004737  032646                  JSR     PC,20$
8521
8522  032566  032777  000002  147044  7$:     BIT     #BIT1,@KMAD0    ;WAIT FOR FIFO DATA
8523  032574  001374                          BNE     7$              ;=1 NO DATA. =0 DATA.
8524  032576  112777  000002  147040          MOVB    #2,@KMAD2       ;READ FIFO.
8525
8526  032604                          8$:
8527  032604  004537  033610                  JSR     R5, $TOUT       ;-TOUT-CHECK FOR TIMEOUT
8528
8529  032610  104000                          ERROR                   ;/TIME-OUT ERROR
8530                                                                  ;/WE FAILED TO cOMPLETE
8531                                                                  ;/CURRENT OPERATION.
8532                                                                  ;/CONTINUES IN THIS LOOP
8533                                                                  ;/WOULD MAKE US "HANG" HERE
8534
8535  032612  000774                          BR              8$
8536
8537                                                                  ;/RETURNS HERE-FROM-TIMED OUT.
8538  032614  122777  000377  147022          CMPB    #377,@KMAD2     ;WAIT FOR READ.
8539  032622  001370                          BNE     8$
8540  032624  105777  147020                  TSTB    @KMAD4          ;WAS A ZERO RETURNED?
8541  032630  001734                          BEQ     6$              ;YES GET NEXT ADDR.
8542                                                                  ;SLAVE WILL RETURN CODE 0 IF
8543  032632  005237  032676                  INC     $AERR           ;DEV PRESENT.   ELSE
8544                                                                  ;EXIT $AERR=1 IF SLAVE GIVES ERROR.
8545  032636  005041                          CLR     -(1)            ;GET RID OF REFERENCE TO BAD ADDR.
8546  032640  012601                  10$:    MOV     (SP)+,R1
8547  032642  000205                          RTS     R5              ;RETURN ALL ADDR. CHECKED.
8548
8549  032644  000000                  11$:    .WORD   0               ;HOLDS DAC CODE PLUS OFFSET
8550                                                                  ;TO SLAVES ADDR. TABLE.
8551
8552  032646  112777  000003  146770  20$:    MOVB    #3,@KMAD2       ;ISSUE FIFO WRITE
8553  032654                          21$:
8554  032654  004537  033610                  JSR     R5, $TOuT       ;-TOUT-CHECK FOR TIMEOUT
8555
8556  032660  104000                          ERROR                   ;/TIME-OUT ERROR
8557                                                                  ;/WE FAILED TO COMPLETE
8558                                                                  ;/CURRENT OPERATION.
8559                                                                  ;/CONTINUES IN THIS LOOP
```

```
8560                                                    ;/WOULD MAKE US "HANG" HERE
8561
8562  032662  000774                      BR          21$
8563
8564                                                    ;/RETURNS HERE-FROM-TIMED OUT.
8565  032664  122777  000377  146752      CMPB    #377,@KMAD2     ;KMC CODE WILL RETURN A "377"
8566  032672  001370                      BNE         21$         ;WHEN DONE COMMAND.
8567  032674  000207                      RTS         PC
8568
8569  032676  000000              $AERR:  .WORD       0           ;=0 IF ADDR. LIST OK,=1 IF BAD.
8570
8571                                      ;*
8572                                      ;*THIS SUB CODE USED TO LOAD MICRO-CODE INTO LPA-11.
8573                                      ;*      CALL =  JSR     R5,$LOAD
8574                                      ;*              .WORD   XX              ;ADDR. OF MICRO CODE.
8575                                      ;*              ;RETURNS HERE
8576                                      ;*      NOTE:   MICRO CODE FILE MUST END IN -1 DATA.
8577                                      ;*
8578
8579  032700  010446              $LOAD:  MOV     R4,-(SP)        ;SAVE R4.
8580  032702  010046                      MOV     R0,-(SP)        ;SAVE R0.
8581  032704  012500              1$:     MOV     (5)+,R0         ;GET PROG. ADDR.
8582  032706  005077  146726              CLR     @KMAD0          ;CLEAR CSR
8583  032712  005077  146732              CLR     @KMAD4          ;CLEAR CRAM ADDR.
8584  032716  052777  002000  146714  2$: BIS     #2000,@KMAD0    ;SELECT CRAM.
8585  032724  012077  146724              MOV     (0)+,@KMAD6     ;WRITE DATA.
8586  032730  052777  020000  146702      BIS     #20000,@KMAD0   ;SET CRAM WRITE
8587  032736  005077  146676              CLR     @KMAD0          ;DISABLE CRAM.
8588  032742  005277  146702              INC     @KMAD4          ;UPDATE CRAM ADDR.
8589  032746  021027  177777              CMP     (0),#-1         ;ALL DONe?
8590  032752  001361                      BNE     2$              ;NO LOOP.
8591  032754  005077  146670              CLR     @KMAD4          ;CLEAR CRAM ADDR.
8592  032760  016500  177776              MOV     -2(5),R0        ;GET MICRO CODE ADDR.
8593
8594  032764  052777  002000  146646  3$: BIS     #2000,@KMAD0    ;SELECT CRAM
8595  032772  022077  146656              CMP     (R0)+,@KMAD6    ;DATA OK?
8596  032776  001013                      BNE     5$              ;NO - REPORT AN ERROR.
8597  033000  021027  177777              CMP     (0),#-1         ;ALL DONE?
8598  033004  001405                      BEQ     4$              ;YES - EXIT
8599  033006  005077  146626              CLR     @KMAD0          ;NO - DESELECT CRAM.
8600  033012  005277  146632              INC     @KMAD4          ;UPDATE CRAM ADDR.
8601  033016  000762                      BR      3$
8602
8603  033020  012600              4$:     MOV     (SP)+,R0        ;RESTORE R0
8604  033022  012604                      MOV     (SP)+,R4        ;RESTORE R4
8605  033024  000205                      RTS     R5              ;EXIT
8606
8607  033026                      5$:                             ;COME HERE ON LOAD ERROR
8608  033026  005745                      TST     -(5)
8609  033030  105204                      INCB    R4              ;UPDATE ERROR COUNTER.
8610  033032  100324                      BPL     1$              ;IF NOT TOO MANY, TRY AGAIN.
8611  033034  000000                      HALT                    ;MICRO CODE LOAD ERROR.
8612                                                              ;KMC-11 FAULT. YOU COULD TRY
8613  033036  000722                      BR      1$              ;TO PRESS CONTINUE TO GIVE IT
```

```
                                                        ;ANOTHER CHANCE, BUT I DOUBT
                                                        ;THAT THAT WOULD WORK. SINCE I'VE
                                                        ;ALREADY GIVEN IT 177 (OCTAL) CHANCES.
                                                        ;TRY RUNNING THE KMC-11 DIAGNOSTIC.


                                                        ;
                                                        ;*THIS ROUTINE ISSUES A WRITE COMMAND TO THE LPA-11
                                                        ;*
                                                        ;*      CALL =  JSR     R5,$TLKW
                                                        ;*              .WORD   0               ;OFFSET OF DEVICE ADDR.
                                                        ;*              .WORD   0               ;DATA TO BE WRITTEN
                                                        ;*
033040  010046                          $TLKW:  MOV     R0,-(SP)        ;SAVE R0
033042  012500                                  MOV     (5)+,R0         ;GET DEVICE OFFSET
033044  052700  000340                          BIS     #340,R0         ;ADD WRITE CODE.
033050  004737  033322                          JSR     PC,$LPW         ;WAIT FOR FAST PATH READY
033054  010037  033146                          MOV     R0,W1
033060  010077  146564                          MOV     R0,@KMAD4
033064  112777  000005  146552                  MOVB    #5,@KMAD2       ;ISSUE FAST PATH WRITE
033072  004737  033322                          JSR     PC,$LPW         ;WAIT FOR RDY
033076  011537  033150                          MOV     (5),W2
033102  112577  146542                          MOVB    (5)+,@KMAD4     ;WRITE LOW BYTE DATA.

033106  112777  000005  146530                  MOVB    #5,@KMAD2       ;FP WRITE
033114  004737  033322                          JSR     PC,$LPW
033120  111537  033152                          MOVB    (5),W3
033124  112577  146520                          MOVB    (5)+,@KMAD4     ;WRITE HIGH BYTE
033130  112777  000005  146506                  MOVB    #5,@KMAD2
033136  004737  033322                          JSR     PC,$LPW
033142  012600                                  MOV     (SP)+,R0
033144  000205                                  RTS     R5              ;EXIT DONE.
033146  000000                          W1:     0
033150  000000                          W2:     0
033152  000000                          W3:     0

                                                        ;*
                                                        ;*THIS ROUTINE ISSUES A READ COMMAND TO THE LPA-11
                                                        ;*
                                                        ;*      CALL =  JSR     R5,$TLKr
                                                        ;*              .WORD   0               ;OFFSET OF DEVICE
                                                        ;*              ;RETURNS HERE
                                                        ;*DATA IN WORD $DATR
                                                        ;*

033154  010046                          $TLKR:  MOV     R0,-(SP)        ;SAVE R0
033156  012500                                  MOV     (5)+,R0         ;GET OFFSET
033160  052700  000300                          BIS     #300,R0         ;ADD READ CODE
033164  004737  033322                          JSR     PC,$LPW         ;WAIT TILL READY
033170  110077  146454                          MOVB    R0,@KMAD4
033174  112777  000005  146442                  MOVB    #5,@KMAD2       ;ISSUE WRITE FP
033202  004737  033322                          JSR     PC,$LPW
033206  010037  033316                          MOV     R0,RD1
033212                                  1$:
```

```
8668   033212  004537  033610              JSR     R5, $TOUt       ;-TOUT-CHECK FOR TIMEOUT
8669
8670   033216  104000                      ERROR                   ;/TIME-OUT ERROR
8671                                                               ;/WE FAILED TO COMPLETE
8672                                                               ;/CURRENT OPERATION.
8673                                                               ;/CONTINUES IN THIS LOOP
8674                                                               ;/WOULD MAKE US "HANG" HERE
8675
8676   033220  000774                      BR              1$
8677
8678                                                               ;/RETURNS HERE-FROM-TIMED OUT.
8679   033222  032777  000040  146410      BIT     #BIT5,@KMADO    ;FAST PATH GOT DATA?
8680   033230  001370                      BNE     1$
8681   033232  112777  000004  146404      MOVB    #4,@KMAD2       ;ISSUE FAST PATH READ
8682   033240  004737  033322              JSR     PC,$LPW
8683   033244  117737  146400  033320      MOVB    @KMAD4,$DATR    ;GET LOW BYTE
8684   033252                          2$:
8685   033252  004537  033610              JSR     R5, $TOUt       ;-TOUT-CHECK FOR TIMEOUT
8686
8687   033256  104000                      ERROR                   ;/TIME-OUT ERROR
8688                                                               ;/WE FAILED TO COMPLETE
8689                                                               ;/CURRENT OPERATION.
8690                                                               ;/CONTINUES IN THIS LOOP
8691                                                               ;/WOULD MAKE US "HANG" HERE
8692
8693   033260  000774                      BR              2$
8694
8695                                                               ;/RETURNS HERE-FROM-TIMED OUT.
8696   033262  032777  000040  146350      BIT     #BIT5,@KMADO    ;FAST PATH READY?
8697   033270  001370                      BNE     2$
8698   033272  112777  000004  146344      MOVB    #4,@KMAD2       ;ISSUE FAST PATH READ
8699   033300  004737  033322              JSR     PC,$LPW
8700   033304  117737  146340  033321      MOVB    @KMAD4,$DATR+1  ;SAVE HIGH BYTE
8701   033312  012600                      MOV     (SP)+,R0
8702   033314  000205                      RTS     R5
8703   033316  000000                  RD1:
8704   033320  000000                  $DATR:  .WORD   0
8705                                           ;
8706                                           ;THIS ROUTINE WAITS FOR KMC-CODE TO BECOME READY AS WELL
8707                                           ;AS FAST PATH TO BE READ.
8708                                           ;
8709                                           ;       CALL =  JSR     PC,$LPW
8710                                           ;
8711                                           ;IT WILL TIME OUT IF TOO MUCH TIME IS TAKEN BY
8712                                           ;THE MICRO-PROCESSORS AND REPORT AN ERROR, THEN HALT.
8713                                           ;
8714
8715   033322  010146                  $LPW:   MOV     R1,-(SP)        ;SAVE R1
8716   033324  005001                          CLR     R1
8717   033326  122777  000377  146310  1$:     CMPB    #377,@KMAD2     ;FINISHED INSTRUCTION?
8718   033334  001403                          BEQ     2$
8719   033336  005201                          INC     R1              ;TIME OUT?
8720   033340  001372                          BNE     1$
8721   033342  000411                          BR      10$
```

```
033344  032777  000020  146266  2$:     BIT     #BIT4,@KMADO    ;FAST PATH READ?
033352  001403                          BEQ     3$
033354  005201                          INC     R1              ;NO - TIME OUT?
033356  001372                          BNE     2$
033360  000402                          BR      10$             ;YES - REPORT AN ERROR

033362  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
033364  000207                          RTS     PC              ;EXIT

033366                          10$:
033366  104401  033374                  TYPE    ,65$            ;;TYPE ASCIZ STRING
033372  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <200>#LPA-11 FAULT#
033412                          64$:

033412  000000                  11$:    HALT                    ;LPA-11 FAULT RUN LPA-11
033414  000776                          BR      11$             ;DIAGNOSTICS.




                                        ;*
                                        ;*THIS ROUTINE PROVIDES THE LINKAGE FROM USER CODE TO
                                        ;*A DEVICE ADDRESS ON THE I/O BUSS FOR WRITE ONLY.
                                        ;*
                                        ;* FIRST WE WILL DETERMINE IF THE ADDRESS HAS BEEN USED
                                        ;* BEFORE. IF NOT WE HAVE TO INITIALIZE THE LPA WITH
                                        ;* THAT ADDRESS.
                                        ;* WHEN THE ADDR. IS KNOWN BY THE LPA, DO THE OUTPUT BY
                                        ;* $TLKW
                                        ;*

033416  010046                  $OUTLP: MOV     R0,-(SP)        ;SAVE R0
033420  010146                          MOV     R1,-(SP)        ;SAVE R1

033422  012700  001666                  MOV     #.DVLS,R0       ;PROGRAM DEFINED LIST.
033426  005001                          CLR     R1
033430  005710                  1$:     TST     (0)             ;TERMINATOR REACHED?
033432  001421                          BEQ     10$             ;YES NEXT STEP.
033434  027520  000000                  CMP     @(5),(0)+       ;MATCH WITH ADDR IN LIST?
033440  001402                          BEQ     2$
033442  005201                          INC     R1
033444  000771                          BR      1$

033446  010137  033464          2$:     MOV     R1,3$           ;SAVE OFFSET, DEVICE KNOWN.
033452  005725                          TST     (5)+
033454  013537  033466                  MOV     @(5)+,4$        ;GET DATA TO BE WRITTEN
033460  004537  033040                  JSR     R5,@TLKW        ;DO WRITE
033464  000000                  3$:     .WORD   0               ;DEVICE OFFSET
033466  000000                  4$:     .WORD   0               ;DATA TO BE WRITTEN.
033470  012601                          MOV     (SP)+,R1
033472  012600                          MOV     (SP)+,R0
033474  000205                          RTS     R5
033476  017520  000000          10$:    MOV     @(5),(0)+       ;SAVE ADDR.
```

```
8776  033502  005010                          CLR     (0)
8777  033504  004537  032020                  JSR     R5,$LPAI
8778  033510  001666                          .WORD   .DVLS
8779  033512  000755                          BR      2$
8780
8781                                          ;*
8782                                          ;*THIS ROUTINE PROVIDES THE LINKAGE FROM USER CODE
8783                                          ;*TO A DEVICE ADDR. ON THE I/O BUSS FOR READ ONLY.
8784                                          ;*
8785                                          ;*FIRST WE WILL DETERMINE IF THE ADDRESS HAS BEEN
8786                                          ;*USED BEFORE. IF NOT, WE HAVE TO INITIALIZE THE LPA
8787                                          ;*WITH THE NEW ADDR.
8788                                          ;*WHEN THE ADDR IS KNOWN WE CAN DO OUTPUT THROUGH
8789                                          ;*$TLKR
8790                                          ;*      CALL THROUGH    MOVEI   DATA,ADDR.
8791                                          ;*              WHICH EQUALS:
8792                                          ;*                      JSR     R5,$INLP
8793                                          ;*                      .WORD   XX      ADDR OF DEVICE
8794                                          ;*                      .WORD   YY      ADDR TO sTORE READ DATA.
8795
8796  033514  010046          $INLP:  MOV     R0,-(SP)        ;SAVE R0
8797  033516  010146                  MOV     R1,-(SP)        ;SAVE R1
8798
8799  033520  012700  001666          MOV     #.DVLS,R0       ;PROG DEFINED ADDR. LIST.
8800  033524  005001                  CLR     R1
8801  033526  005710          1$:     TST     (0)             ;EOL REACHED?
8802  033530  001420                  BEQ     10$             ;YES - DEFINE NEW ADDR.
8803
8804  033532  027520  000000          CMP     @(5),(0)+       ;ADDR. MATCH?
8805  033536  001402                  BEQ     2$
8806  033540  005201                  INC     R1
8807  033542  000771                  BR      1$
8808
8809  033544  010137  033556  2$:     MOV     R1,3$           ;SAVE LIST OFFSET
8810  033550  005725                  TST     (5)+
8811  033552  004537  033154          JSR     R5,$TLKR        ;GO READ DEVICE
8812          033556          $OFS=.
8813  033556  000000          3$:     .WORD   0               ;OFFSET OF DEVICE
8814
8815  033560  013735  033320          MOV     $DATR,@(5)+     ;STORE DATA.
8816  033564  012601                  MOV     (SP)+,R1        ;RESTORE R1
8817  033566  012600                  MOV     (SP)+,R0        ;RESTORE R2
8818  033570  000205                  RTS     R5              ;EXIT
8819
8820  033572  017520  000000  10$:     MOV     @(5),(0)+
8821  033576  005010                  CLR     (0)
8822  033600  004537  032020          JSR     R5,$LPAI
8823  033604  001666                  .WORD   .DVLS
8824  033606  000756                  BR      2$
8825                                          ;*
8826                                          ;*$TOUT ROUTINE USED TO WATCH IF
8827                                          ;*      WE'RE IN A LOOP TOO-LONG
8828                                          ;*      CALL=   JSR R5, $TOUT
8829                                          ;*              ERROR X     ;RETURNS HERE ON TIMEOUT
```

```
8830                                                    ;*              BR
8831                                                    ;*              ;RETURNS HERE NO ERROR
8832                                                    ;*
8833
8834  033610  020537  033644          $TOUT:  CMP     R5,$SAD         ;SAME ADDR?
8835  033614  001405                  BEQ     1$
8836  033616  010537  033644          MOV     R5,$SAD         ;NO-SAVE THIS ADDR.
8837  033622  005037  033646          CLR     $CNT            ;CLR CNT AT ADDR.
8838  033626  000403                  BR      2$
8839  033630  005237  033646          1$:     INC     $CNT            ;OVERFLOW?
8840  033634  100402                  BMI     3$              ;YES-ERROR RETURN
8841  033636  062705  000004          2$:     ADD     #4,R5           ;NO-NON ERROR RETURN
8842  033642  000205                  3$:     RTS     R5              ;RETURN.
8843
8844  033644  000000          $SAD:   .WORD   0               ;CONTAINS LOOP ADDR.
8845  033646  000000          $CNT:   .WORD   0               ;# OF TIMES AT ADDR.
8846
8847                                                    ;*
8848                                                    ;* THIS ROUTINE REPLACES WHAT THE USER WOULD ORDINARILY
8849                                                    ;*USE FOR A RESET.  FIRST,WE DO A RESET INSTRUCTION.
8850                                                    ;*THEN WE CLR ".DVLST" WHICH FORCES US TO rESET BOTH THE
8851                                                    ;*KMC AND DMC AS SOON AS A DEVICE IS REFERENCED.
8852                                                    ;*
8853                                                    ;*      CALL=JSR        PC,$RESET       ;REPLACES "RESET INSTRUCTION
8854                                                    ;*              ;RETURNS HERE.
8855                                                    ;*
8856  033650  000005          $RESET: RESET                           ;RESET THE WORLD.
8857
8858                                  ;*        MOV     @2$,1$  ;/READ DEVICE REG 2$,PUT DATA IN 1$.
8859  033662  005737  032676                  TST     $AERR           ;IF NO ERROR,LOOP
8860  033666  001004                  BNE     10$             ;THERE WAS AN ERROR.
8861  033670  062737  000002  033704          ADD     #2,2$           ;UPDATE DEVICE ADDR.
8862                                                                  ;YOU SEE ,WE HAVE TO PROTECT OUR SELF!
8863                                                                  ;IF 2$ CONTAINED A VALID ADDR,WE
8864                                                                  ;MUST KEEP TRYING UNTIL WE GENERATE
8865                                                                  ;AN INVALID ADDR.
8866  033676  000764                  BR      $RESET
8867  033700                  10$:
8868  033700  000207                  RTS     PC
8869  033702  000000          1$:     .WORD   0               ;JUNK LOC.
8870  033704  160000          2$:     .WORD   160000          ;DUMB ADDR. FORCES INIT OF DMC/KMC.
8871
8872
8873                                                    ;
8874                                                    ;SDELAY- ROUTINE TO GIVE A MINOR DELAY.
8875                                                    ;        IS NOT TIME DEPENDENT CODE SENCE
8876                                                    ;        NOT USED TO GET SPECIFIC TIME BUT
8877                                                    ;        JUST A LITTLE DELAY.
8878
8879                                                    ;       THAT IS UNLESS A REAL TIME CLOCK IS PRESENT!
8880                                                    ;       THEN WE'LL GENERATE A TIME BETWEEN 16MS TO 32 MS
8881                                                    ;
8882                                                    ;
8883                                                    ;       CALL=   JSR PC,  SDELAY
```

```
8884                                            ;
8885  033706                            SDELAY:
8886  033706  005737  033770                     TST     RTCCSR          ;CLOCK PRESENT?
8887  033712  100016                             BPL     10$
8888  033714  012737  000002  033760             MOV     #2,TIME
8889  033722  052777  000115  000040             BIS     #115,@RTCCSR    ;START CLOCK
8890  033730  005037  177776                     CLR     PS
8891  033734  005737  033760            1$:      TST     TIME
8892  033740  001375                             BNE     1$
8893  033742  005077  000022                     CLR     @RTCCSR         ;STOP CLOCK
8894
8895  033746  000207                             RTS PC
8896  033750  105237  033760            10$:     INCB    TIME
8897  033754  001375                             BNE     10$
8898  033756  000207                             RTS     PC
8899
8900  033760  000000                    TIME:    .WORD   0
8901
8902  033762  005337  033760            CLKINT:  DEC     TIME
8903  033766  000002                             RTI
8904  033770  000000                    RTCCSR:  .WORD   0               ;CLOCK CSR IF USED.
8905                                            ;
8906                                            ;*
8907                                            ;*THIS MACRO ALLOWS THE OPERATOR TO TALK TO
8908                                            ;*ANY DEVICE ON THE I/O BUS
8909                                            ;*USER MUST START AT THIS ADDR.
8910                                            ;*HE MUST SAY EITHER "E" FOR EXAMINE, OR "D" FOR DEPOSIT.
8911                                            ;*"E" IS DEFAULT.
8912                                            ;*NEXT, HE MUST SUPPLY AN ADDR.
8913                                            ;*NOTE IF ADDR. IS NOT FOUND ON I/O BUS, A HALT
8914                                            ;*WILL OCCUR.
8915
8916  033772                            $UTK:
8917  033772  005037  001666                     CLR     .DVLS
8918  033776                            21$:
8919  033776  104401  034004                     TYPE    ,65$            ;;TYPE ASCIZ STRING
8920  034002  000405                             BR      64$             ;;GET OVER THE ASCIZ
8921                                    ;;65$:   .ASCIZ  <200>#E OR D?#
8922  034016                            64$:
8923  034016  105777  145122            1$:      TSTB    @$TKS
8924  034022  100375                             BPL     1$
8925  034024  117737  145116  034146             MOVB    @$TKB,20$       ;GET INPUT
8926  034032  104401  034146                     TYPE,   20$             ;ECHO, NEXT MESSAGE.
8927  034036  142737  000240  034146             BICB    #240,20$        ;STRIP PARITY, LC
8928  034044  104412                             RDOCT                   ;GET ADDR.
8929  034046  012637  034144                     MOV     (SP)+,14$
8930  034052  123727  034146  000104             CMPB    20$,#'D         ;DEPOSIT?
8931  034060  001411                             BEQ     10$
8932
8933  034062  004537  033514                     JSR     R5,$INLP        ;GET DATA
8934  034066  034144                    2$:      .WORD   14$
8935  034070  034102                             .WORD   5$
8936
8937  034072  013746  034102                     MOV     5$,-(SP)        ;;SAVE 5$ FOR TYPEOUT
```

```
                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                        BR      21$             ;LOOP.
                                5$:     .WORD   0
                                10$:
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <200>#DATA= #
                                66$:
                                        RDOCT
                                        MOV     (SP)+,13$
                                11$:    JSR     R5,$OUTLP       ;OUTPUT ROUTINE.
                                12$:    .WORD   14$             ;DEVICE ADDR.
                                        .WORD   13$             ;DATA
                                        BR      21$
                                13$:    .WORD   0
                                14$:    .WORD   0
                                20$:    .ASCIZ  <1><200>#DEVICE ADDR= #
                                        .EVEN
;
;THIS ROUTINE LOOKS THROUGH CURENT .DVLS FOR A/D ADDR.
;IF UNFOUND,GENERATES IT. THIS ROUTINE'S WHOLE PURPOSE IS
;TO SET UP THE USER PROGRAM TO LINK TO FILE "DRLPX2" FOR
;SAMPLE TAKEING PURPOSES.
;       TO TAKE SAMPLES, THE USER PROGRAM MUST SET UP
;       A/D CSR IN BSEL 4,AND 5.
;       (2) HE MUST CALL THIS ROUTINE:
;               JSR     R5,$PUTS        ;CALL SET UP ROUTINE.
;               .WORD   ADCSR           ;ADDR. OF A/D CSR.
;               ;RETURNS HERE ;KMC BSEL 3,6,7 PERMINENTLY SET UP
;                               ;(UNTILL ONE DOES A RESET)
;
;       (3)THE USER MUST PUT CODE 006 INTO KMC REG 2 TO
;          START CONVERSION  CAUTION*DO wITH MOVB INSTR.!
;       (4)MONITOR KMC REG 2 FOR CODE 377 (DRLPX2 IS DONE)
;       (5)READ KMC REG 4,5 FOR A/D RESULT.
;       (6)  TO TAKE MORE SAMPLES,SIMPLY PUT A/D CSR INTO
;          BSEL 4,5 AND CODE 6 INTO BSEL 2.
;
$PUTS:  MOV     (5)+,1$                 ;GET ADDR OF ADDR. OF A/D
        JSR     R5,$INLP
1$:     .WORD   0
        .WORD   10$
        MOVB    $OFS,@KMAD6
        MOVB    $OFS,@KMAD7
        MOV     1$,2$
        ADD     #2,2$
```

```
 8992  034232  004537  033514            JSR     R5,$INLP
 8993  034236  000000            2$:     .WORD   0
 8994  034240  034274                    .WORD   10$
 8995  034242  113777  033556  145376    MOVB    $OFS,@KMAD3
 8996  034250  152777  000340  145376    BISB    #340,@KMAD6
 8997  034256  152777  000300  145372    BISB    #300,@KMAD7
 8998  034264  152777  000300  145354    BISB    #300,@KMAD3
 8999  034272  000205                    RTS     R5
 9000  034274  000000            10$:    .WORD   0
 9001
 9002
 9003          000001                    .END
```

```
ABASE = 170404          220#    317     358
ABR     001330          384#    708*    761     805     807     844     846    1664    1666    1684    1686    1719    1721
                       1739    1741    1774    1776    1794    1796    1829    1831    1849    1851    1884    1886    1904
                       1906    1939    1941    1959    1961    1994    1996    2014    2016    2049    2051    2069    2071
                       2104    2106    2124    2126    2159    2161    2179    2181    2214    2216    2234    2236    2269
                       2271    2289    2291    2324    2326    2344    2346    2379    2381    2399    2401    2434    2436
                       2454    2456    2489    2491    2509    2511    3498    3548    3598    3878    3946    4020    4094
                       4166    4219    4272    4325    4378    4431    4485    4546    4554    4601    4656    4780    4786
                       4796    4849    4855    4865    4907    4955    5008    5077    5095    5166    5902    5983    6046
                       6157    6268    6379    7111    7299    8381
ACDW1 = 000000          317     360
ACDW2 = 000000          317     361
ACPUOP= 000000          317     332
ACR     001332          385#    710*    763    3503    3550    3600    3894    3968    4123    4175    4228    4281    4334
                       4387    4440    4495    4559    4666    4917    4965    5015    5116    5175    5507    5921    5993
                       6067    6081    6109    6178    6192    6220    6289    6303    6331    6400    6414    6442    7123
                       8383    8395    8397
ADDW0 = 000000          317     362
ADDW1 = 000000          317     363
ADDW10= 000000          317     372
ADDW11= 000000          317     373
ADDW12= 000000          317     374
ADDW13= 000000          317     375
ADDW14= 000000          317     376
ADDW15= 000000          317     377
ADDW2 = 000000          317     364
ADDW3 = 000000          317     365
ADDW4 = 000000          317     366
ADDW5 = 000000          317     367
ADDW6 = 000000          317     368
ADDW7 = 000000          317     369
ADDW8 = 000000          317     370
ADDW9 = 000000          317     371
ADEVCT= 000000          317     323
ADEVM = 000000          317     359
AENV  = 000000          317     328
AENVM = 000000          317     329
AFATAL= 000000          317     320
AMADR1= 000000          317     345
AMADR2= 000000          317     349
AMADR3= 000000          317     352
AMADR4= 000000          317     355
AMAMS1= 000000          317     339
AMAMS2= 000000          317     347
AMAMS3= 000000          317     350
AMAMS4= 000000          317     353
AMSGAD= 000000          317     325
AMSGLG= 000000          317     326
AMSGTY= 000000          317     319
AMTYP1= 000000          317     340
AMTYP2= 000000          317     348
AMTYP3= 000000          317     351
AMTYP4= 000000          317     354
APASS = 000000          317     322
```

```
APRIOR= 000006   222#   317
APRITY  001352   397#   726
APTCSU= 000040  8027   8132#
APTENV= 000001  7623   8020   8088   8130#
APTSIZ= 000200   669   8129#
APTSPO= 000100  8022   8090   8131#
ASR     001326   383#   681*   706   759   869   891   908  1004  1006  1024  1026  1059  1061
                1079  1081  1114  1116  1134  1136  1169  1171  1189  1191  1224  1226  1244
                1246  1279  1281  1299  1301  1334  1336  1354  1356  1389  1391  1409  1411
                1444  1446  1464  1466  1499  1501  1519  1521  1554  1556  1574  1576  1609
                1611  1629  1631  3496  3542  3588  3781  3783  3788  3792  3827  3829  3832
                3834  3853  3876  3885  3887  3890  3944  3954  3959  3962  4017  4025  4027
                4030  4034  4051  4085  4097  4099  4102  4106  4164  4169  4181  4217  4222
                4234  4270  4275  4287  4323  4328  4340  4376  4381  4393  4429  4434  4446
                4483  4488  4500  4544  4549  4598  4604  4606  4609  4611  4653  4659  4661
                4664  4714  4722  4724  4727  4730  4777  4783  4788  4791  4846  4852  4857
                4860  4904  4910  4912  4915  4952  4958  4960  4963  5005  5010  5013  5074
                5083  5085  5088  5160  5169  5179  5195  5239  5242  5295  5321  5324  5383
                5396  5399  5452  5492  5499  5545  5602  5659  5716  5773  5830  5898  5914
                5916  5919  5980  5985  5988  6044  6051  6054  6062  6065  6100  6103  6155
                6162  6165  6173  6176  6211  6214  6266  6273  6276  6284  6287  6322  6325
                6377  6384  6387  6395  6398  6433  6436  6490  6509  6512  6550  6553  6604
                6623  6626  6664  6667  6718  6737  6740  6778  6781  6832  6851  6854  6892
                6895  6999  7013  7015  7018  7021  7096  7115  7117  7120  7199  7206  7209
                7221  7292  7304  7308  7311  7322  7394  7404  7426  8379  8393
ASWREG= 000000   317   330
ATESTN= 000000   317   321
AUNIT = 000000   317   324
.USWR = 000000   317   331
AVECP2  001344   392#   720*
AVECT   001342   391#   718
AVECT1= 000344   221#   317   356   620   621
AVECT2= 000000   317   357
BBR     001336   388#   714*   767   965   967  3049  3051  3069  3071  3104  3106  3124  3126
                3159  3161  3179  3181  3214  3216  3234  3236  3269  3271  3289  3291  3324
                3326  3344  3346  3379  3381  3399  3401  3434  3436  3454  3456  3644  3693
                3743  5229  5306  5388  5457  5496  5549  5606  5663  5720  5777  5834  5905
                6492  6606  6720  6834  7401  8387
BCR     001340   389#   716*   769  3649  3697  3747  5244  5329  5417  5464  5560  5617  5674
                5731  5788  5845  6514  6529  6558  6628  6643  6672  6742  6757  6786  6856
                6871  6900  8389  8399  8401
BIT0  = 000001   204#  1606  1661  2991  3046  5234  5977
BIT00 = 000001   194#   204  3834  4106  4611
BIT01 = 000002   193#   203
BIT02 = 000004   192#   202
BIT03 = 000010   191#   201
BIT04 = 000020   190#   200
BIT05 = 000040   189#   199  4034  4181  4234  4287  4340  4393  4446  4500  7308
BIT06 = 000100   188#   198
BIT07 = 000200   187#   197  4051  4730  7414
BIT08 = 000400   186#   196  7786
BIT09 = 001000   185#   195  7634  7794
BIT1  = 000002   203#  1551  1716  2935  3101  5234  8522
BIT10 = 002000   184#  2211  4724  4788  4857  4912  4960  5010  7015  7611
```

```
BIT11 = 004000   183#  2266  2543  5160  5221  5239  5296  5321  5396  5916  6046  6062  6100
                 6157  6173  6211  6268  6284  6322  6379  6395  6433  6492  6509  6550  6606
                 6623  6664  6720  6737  6778  6834  6851  6892  7801  8462
BIT12 = 010000   182#  2321  3785  3829  3887  3959  4027  4099  4606  4661  5085  5985  7117
                 7206
BIT13 = 020000   181#  1111  2376  3824  7036  7138  7235  7337  7441  7618
BIT14 = 040000   180#  1056  2431  3987  5348  7772  8454
BIT15 = 100000   179#  1001  2486  7018  8462
BIT2  = 000004   202#  1496  1771  2879  3156
BIT3  = 000010   201#  1441  1826  2823  3211
BIT4  = 000020   200#  1881  2767  3266  8723
BIT5  = 000040   199#  1386  1936  2711  3321  5234  6492  6606  6720  6834  8466  8679  8696
BIT6  = 000100   198#  1331  1991  2655  3376
BIT7  = 000200   197#  1276  2046  2599  3431
BIT8  = 000400   196#  1221  2101  7112
BIT9  = 001000   195#  1166  2156  7112  7206
BPRITY  001354   398#   726*
BPTVEC= 000014   211#
BSR     001334   387#   712*   765   928   930  2546  2548  2566  2568  2602  2604  2622  2624
                2658  2660  2678  2680  2714  2716  2734  2736  2770  2772  2790  2792  2826
                2828  2846  2848  2882  2884  2902  2904  2938  2940  2958  2960  2994  2996
                3014  3016  3642  3685  3735  4540  4709  4768  4837  5072  5080  5137  5163
                5224  5234  5237  5299  5312  5315  5385  5391  5394  5401  5454  5494  5547
                5552  5565  5604  5609  5622  5661  5666  5679  5718  5723  5736  5775  5780
                5793  5832  5837  5850  5900  5908  5911  6042  6049  6153  6160  6264  6271
                6375  6382  6488  6495  6497  6500  6503  6602  6609  6611  6614  6617  6716
                6723  6725  6728  6731  6830  6837  6839  6842  6845  7001  7098  7201  7294
                7396  7411  7414  8385  8394
BVECT   001346   394#   722*
BVECT2  001350   395#   724*
CKSWR = 104407  7607  7633  7771  8215#
CLKINT  033762  8902#
CR    = 000015   119#  8066  8076
CRLF  = 000200   120#  8037  8076
DDISP = 177570   126#   294   657
DF0     032014   423   431   439   447   455   463   471   480   488   496   501   502   503
                 504   511   519   527   535   542   550   558   566   574   582   590  8405#
DH1     030777   421   429  8301#
DH10    031263   477  8337#
DH12    031422   494   525  8354#
DH14    031441   509   533  8357#
DH20    031457   540  8360#
DH23    031521   564  8366#
DH26    031563   588  8372#
DH3     031035   437  8307#
DH4     031073   445   486   548   556  8313#
DH5     031131   453  8319#
DH6     031167   461  8325#
DH7     031225   469   517   572   580  8331#
DISPLA  001142   294#   657*  665*  7610*  7816*
DISPRE  000174    97#   665
DRLPX2= ******  G  14#  8452
DSWR  = 177570   125#   293   656
DT1     031612   422   430   541   565  8379#
```

```
DT10     031706              479    8391#
DT12     031722              495     526    8393#
DT14     031730              510     534    8394#
DT21     031736              549    8395#
DT22     031750              557    8397#
DT24     031762              573    8399#
DT25     031774              581    8401#
DT26     032006              589    8403#
DT3      031624              438    8381#
DT4      031636              446     487    8383#
DT5      031650              454    8385#
DT6      031662              462    8387#
DT7      031674              470     518    8389#
EMTVEC=  000030              214#    641*    642*
EM1      030112              420    8220#
EM10     030357              476    8249#
EM11     030404              485     547     555    8253#
EM12     030433              493    8257#
EM14     030473              508    8263#
EM15     030533              516     571     579    8269#
EM16     030562              524    8273#
EM17     030615              532    8278#
EM2      030145              428    8225#
EM20     030650              539    8283#
EM23     030707              563    8289#
EM26     030746              587    8295#
EM3      030174              436    8229#
EM4      030223              444    8233#
EM5      030252              452    8237#
EM6      030301              460    8241#
EM7      030330              468    8245#
ERRVEC=  000004              207#    654    655*    666*   7777    7778*   7780*   7783*
GNS   =  ****** U             96     678   7045    7052    7147    7154    7244    7251    7346    7353    7450    7457    7472
                            7480    8207   8208    8209    8210    8211    8213    8215    8216    8217    8218    8219    8445
                            8499    8503   8507    8735    8921    8945
GTSWR =  104406             8213#
HT    =  000011              117#    8035   8076
IOTRD    024612             7467#   8219
IOTT  =  104413             8219#
IOTVEC=  000020              212#    639*    640*
KMAD0    001640              603#    691   8454*   8462*   8466    8522    8582*   8584*   8586*   8587*   8594*   8599*   8679
                            8696    8723
KMAD1    001642              606#    692
KMAD2    001644              608#   8471*   8484    8524*   8538    8552*   8565    8633*   8638*   8642*   8664*   8681*   8698*
                            8717
KMAD3    001646              610#   8995*   8998*
KMAD4    001650              612#   8486    8491    8515*   8517*   8519*   8540    8583*   8588*   8591*   8600*   8632*   8636*
                            8641*   8663*   8683    8700
KMAD5    001652              614#
KMAD6    001654              616#   8585*   8595    8988*   8996*
KMAD7    001656              618#    696    8989*   8997*
LF    =  000012              118#   8070    8076
LOOP     002310              702#   6951
LPADH    001652              613#
```

```
LPADL   001650            611#
LPCI    001640            602#
LPCO    001644            607#
LPMR    001642            605#
LPMS1   001654            615#
LPMS2   001656            617#
LPSO    001646            609#
LS210   023376            103    6986#
LS214   023564            104    7083#
LS220   024000            105    7186#
LS224   024160            106    7279#
LS230   024372            107    7381#
P     = 000012           1644#  1699#  1754#  1809#  1864#  1919#  1974#  2029#  2084#  2139#  2194#  2249#  2304#
                         2359#  2414#  2469#  2524#  3029#  3084#  3139#  3194#  3249#  3304#  3359#  3414#  3469#
                         4811#  4880#
PC    =%000007            138#  5458*  6935*  6938*  6945*  6950   7620*  7626*  7679*  7912*  8025*  8044*  8051*
                         8058*  8072*  8074*  8107*  8124*  8516*  8518*  8520*  8567*  8630*  8634*  8639*  8643*
                         8662*  8665*  8682*  8699*  8730*  8868*  8895*  8898*
PIRQ  = 177772            124#
PIRQVE= 000240            218#
PR0   = 000000            141#
PR1   = 000040            142#
PR2   = 000100            143#
PR3   = 000140            144#
PR4   = 000200            145#
PR5   = 000240            146#
PR6   = 000300            147#
PR7   = 000340            148#
PS    = 177776            121#    122   8890*
PSW   = 177776            122#
PWRVEC= 000024            213#    645*   646*  8137*  8138*  8147*  8153*  8165*  8166*
RDCHR = 104410           7971    8216#
RDLIN = 104411           7835    8217#
RDOCT = 104412           8218#   8928   8947
RD1     033316           8666*  8703#
RESVEC= 000010            208#
RSTART  002434            102     727#
RTCCSR  033770           8886    8889*  8893*  8904#
R0    =%000000            129#    689    691*   694*   695    700*   703*   704*   706*   707*   708    709*   710
                          711*   712    713*   714    715*   716    718*   719*   720    721*   722    723*   724
                         4169*  4170*  4222*  4223*  4275*  4276*  4328*  4329*  4381*  4382*  4434*  4435*  4489*
                         4490*  5169*  5170*  5499*  5500*  5553*  5554*  5610*  5611*  5667*  5668*  5724*  5725*
                         5781*  5782*  5838*  5839*  6056*  6073*  6095*  6103*  6167*  6184*  6206*  6214*  6278*
                         6295*  6317*  6325*  6389*  6406*  6428*  6436*  6503*  6521*  6545*  6553*  6617*  6635*
                         6659*  6667*  6731*  6749*  6773*  6781*  6845*  6863*  6887*  6895*  6942*  6945   7651
                         7652*  7653*  7660*  7661*  7662*  7663*  7664*  7665   7670   7675*  7677*  7681   7683
                         7702   7712*  7716   7732   7733   7746*  7832   7836*  7839   7855*  8018   8019*  8024
                         8029   8032*  8084   8092*  8096   8097   8099*  8100*  8101   8123*  8139   8164*  8184
                         8185*  8186   8187*  8188*  8189*  8190*  8580   8581*  8592*  8595   8603*  8627   8628*
                         8629*  8631   8632   8644*  8659   8660*  8661*  8663   8666   8701*  8754   8757*  8773*
                         8796   8799*  8817*
R1    =%000001            130#    690    692*   696    699*   7703   7716*  7717   7721   7745*  7833   7837*  7841*
                         7843*  7845*  7848*  7851   7854*  8085   8122*  8140   8163*  8458   8459*  8460*  8463*
                         8511*  8512   8517   8519   8546*  8715   8716*  8719*  8725*  8729*  8755   8758*  8763*
```

```
                    8766    8772*   8797    8800*   8806*   8809    8816*
R2      =%000002     131#    7704    7715*   7719*   7722    7729*   7730*   7731    7736*   7744*   7834    7838*   7842*
                    7844*   7846*   7852    7853*   8141    8162*
R3      =%000003     132#    7548    7557*   7563*   7564*   7567*   7572*   7573*   7574    7583*   7705    7713*   7714*
                    7728*   7731*   7740*   7741*   7743*   7967    7968*   7969    7972*   7973    7977    7979    7981*
                    7983*   8142    8161*
R4      =%000004     133#    7549    7551*   7552*   7553*   7554    7555*   7569    7571*   7579*   7582*   8143    8160*
                    8579    8604*   8609*
R5      =%000005     134#     759*    761*    763*    765*    767*    769*    805*    807*    844*    846*    889*    891*
                     908*    928*    930*    965*    967*   1004*   1006*   1024*   1026*   1059*   1061*   1079*   1081*
                    1114*   1116*   1134*   1136*   1169*   1171*   1189*   1191*   1224*   1226*   1244*   1246*   1279*
                    1281*   1299*   1301*   1334*   1336*   1354*   1356*   1389*   1391*   1409*   1411*   1444*   1446*
                    1464*   1466*   1499*   1501*   1519*   1521*   1554*   1556*   1574*   1576*   1609*   1611*   1629*
                    1631*   1664*   1666*   1684*   1686*   1719*   1721*   1739*   1741*   1774*   1776*   1794*   1796*
                    1829*   1831*   1849*   1851*   1884*   1886*   1904*   1906*   1939*   1941*   1959*   1961*   1994*
                    1996*   2014*   2016*   2049*   2051*   2069*   2071*   2104*   2106*   2124*   2126*   2159*   2161*
                    2179*   2181*   2214*   2216*   2234*   2236*   2269*   2271*   2289*   2291*   2324*   2326*   2344*
                    2346*   2379*   2381*   2399*   2401*   2434*   2436*   2454*   2456*   2489*   2491*   2509*   2511*
                    2546*   2548*   2566*   2568*   2602*   2604*   2622*   2624*   2658*   2660*   2678*   2680*   2714*
                    2716*   2734*   2736*   2770*   2772*   2790*   2792*   2826*   2828*   2846*   2848*   2882*   2884*
                    2902*   2904*   2938*   2940*   2958*   2960*   2994*   2996*   3014*   3016*   3049*   3051*   3069*
                    3071*   3104*   3106*   3124*   3126*   3159*   3161*   3179*   3181*   3214*   3216*   3234*   3236*
                    3269*   3271*   3289*   3291*   3324*   3326*   3344*   3346*   3379*   3381*   3399*   3401*   3434*
                    3436*   3454*   3456*   3496*   3498*   3503*   3542*   3548*   3550*   3588*   3598*   3600*   3642*
                    3644*   3649*   3685*   3693*   3697*   3735*   3743*   3747*   3781*   3783*   3788*   3792*   3827*
                    3829*   3832*   3834*   3853*   3876*   3878*   3885*   3887*   3890*   3894*   3944*   3946*   3954*
                    3959*   3962*   3968*   4017*   4020*   4025*   4027*   4030*   4034*   4051*   4085*   4094*   4097*
                    4099*   4102*   4106*   4123*   4164*   4166*   4169*   4175*   4181*   4217*   4219*   4222*   4228*
                    4234*   4270*   4272*   4275*   4281*   4287*   4323*   4325*   4328*   4334*   4340*   4376*   4378*
                    4381*   4387*   4393*   4429*   4431*   4434*   4440*   4446*   4483*   4485*   4488*   4495*   4500*
                    4540*   4544*   4546*   4549*   4554*   4559*   4598*   4601*   4604*   4606*   4609*   4611*   4653*
                    4656*   4659*   4661*   4664*   4666*   4709*   4714*   4722*   4724*   4727*   4730*   4768*   4777*
                    4780*   4783*   4786*   4788*   4791*   4796*   4837*   4846*   4849*   4852*   4855*   4857*   4860*
                    4865*   4904*   4907*   4910*   4912*   4915*   4917*   4952*   4955*   4958*   4960*   4963*   4965*
                    5005*   5008*   5010*   5013*   5015*   5072*   5074*   5077*   5080*   5083*   5085*   5088*   5095*
                    5116*   5137*   5160*   5163*   5166*   5169*   5175*   5179*   5195*   5224*   5229*   5234*   5237*
                    5239*   5242*   5244*   5295*   5299*   5306*   5312*   5315*   5321*   5324*   5329*   5383*   5385*
                    5388*   5391*   5394*   5396*   5399*   5401*   5417*   5452*   5454*   5457*   5464*   5492*   5494*
                    5496*   5499*   5507*   5545*   5547*   5549*   5552*   5560*   5565*   5602*   5604*   5606*   5609*
                    5617*   5622*   5659*   5661*   5663*   5666*   5674*   5679*   5716*   5718*   5720*   5723*   5731*
                    5736*   5773*   5775*   5777*   5780*   5788*   5793*   5830*   5832*   5834*   5837*   5845*   5850*
                    5898*   5900*   5902*   5905*   5908*   5911*   5914*   5916*   5919*   5921*   5980*   5983*   5985*
                    5988*   5993*   6042*   6044*   6046*   6049*   6051*   6054*   6062*   6065*   6067*   6081*   6100*
                    6103*   6109*   6153*   6155*   6157*   6160*   6162*   6165*   6173*   6176*   6178*   6192*   6211*
                    6214*   6220*   6264*   6266*   6268*   6271*   6273*   6276*   6284*   6287*   6289*   6303*   6322*
                    6325*   6331*   6375*   6377*   6379*   6382*   6384*   6387*   6395*   6398*   6400*   6414*   6433*
                    6436*   6442*   6488*   6490*   6492*   6495*   6497*   6500*   6503*   6509*   6512*   6514*   6529*
                    6550*   6553*   6558*   6602*   6604*   6606*   6609*   6611*   6614*   6617*   6623*   6626*   6628*
                    6643*   6664*   6667*   6672*   6716*   6718*   6720*   6723*   6725*   6728*   6731*   6737*   6740*
                    6742*   6757*   6778*   6781*   6786*   6830*   6832*   6834*   6837*   6839*   6842*   6845*   6851*
                    6854*   6856*   6871*   6892*   6895*   6900*   6999*   7001*   7013*   7015*   7018*   7021*   7096*
                    7098*   7111*   7115*   7117*   7120*   7123*   7199*   7201*   7206*   7209*   7221*   7292*   7294*
                    7299*   7304*   7308*   7311*   7322*   7394*   7396*   7401*   7404*   7411*   7414*   7426*   7550
                    7556*   7558*   7560*   7561*   7562*   7563    7581*   7706    7708*   7710*   7717*   7721*   7736
```

```
                    7742*  8144   8159*  8451*  8473*  8527*  8547*  8554*  8605*  8645*  8668*  8685*  8702*
                    8769*  8774*  8777*  8811*  8818*  8822*  8834   8836   8841*  8842*  8859*  8933*  8950*
                    8985*  8992*  8999*
R6      =%000006     135*   633*   634*   635   7468   7483*  7485
R7      =%000007     136*
SDELAY   033706     8885*
SP      =%000006     137*   637*   654*   662*   666    689*   690*   699    700    727*   728*   729*   7475*
                    7487*  7540*  7541   7542   7543*  7548*  7549*  7550*  7556   7581   7582   7583   7584*
                    7585*  7615   7636*  7639*  7651*  7656*  7677   7681*  7702*  7703*  7704*  7705*  7706*
                    7707*  7708   7711*  7724   7726*  7728   7738   7740   7742   7743   7744   7745   7746
                    7748*  7749*  7777*  7780   7782   7783   7812   7813   7817*  7830*  7831*  7832*  7833*
                    7834*  7836   7839*  7847*  7848   7850   7851*  7853   7854   7855   7872*  7873*  7874
                    7881*  7884*  7885*  7889*  7890*  7894   7897*  7901   7903   7905   7906*  7913   7915
                    7917*  7918   7920*  7921*  7922*  7923*  7924*  7939*  7940*  7943*  7944*  7945   7949*
                    7950*  7951   7954   7956   7958*  7967*  7972   7983   7984*  7985*  7986*  8018*  8019
                    8029*  8031   8032   8033*  8035   8037   8039   8045   8047*  8049*  8057*  8061   8065
                    8066   8070   8084*  8085*  8092   8093*  8104   8105*  8106*  8116   8117*  8122   8123
                    8139*  8140*  8141*  8142*  8143*  8144*  8145*  8146   8154*  8158   8159   8160   8161
                    8162   8163   8164   8184*  8185   8195*  8196*  8426*  8448   8449   8458*  8546   8579*
                    8580*  8603   8604   8627*  8644   8659*  8701   8715*  8729   8754*  8755*  8772   8773
                    8796*  8797*  8816   8817   8929   8937*  8948
STACK = 001100       112*   637    727
START   001730       100    630*
STKLMT= 177774       123*
SWR     001140       293*   635    656*   658    664*   671*   3987   5348   7036   7138   7235   7337   7441
                    7611   7618   7630   7634   7772   7786   7788   7794   7801   7868   7905*  8145   8158*
SWREG   000176        98*   664    7868   7881
SW0     = 000001     176*
SW00    = 000001     166*   176
SW01    = 000002     165*   175
SW02    = 000004     164*   174
SW03    = 000010     163*   173
SW04    = 000020     162*   172
SW05    = 000040     161*   171
SW06    = 000100     160*   170
SW07    = 000200     159*   169
SW08    = 000400     158*   168
SW09    = 001000     157*   167
SW1     = 000002     175*
SW10    = 002000     156*
SW11    = 004000     155*
SW12    = 010000     154*
SW13    = 020000     153*
SW14    = 040000     152*
SW15    = 100000     151*
SW2     = 000004     174*
SW3     = 000010     173*
SW4     = 000020     172*
SW5     = 000040     171*
SW6     = 000100     170*
SW7     = 000200     169*
SW8     = 000400     168*
SW9     = 001000     167*
TBITVE= 000014       209*
```

```
TIME    033760          8888*   8891    8896*   8900*   8902*
TKVEC = 000060           216#
TPVEC = 000064           217#
TRAPVE= 000034           215#    643*    644*
TRTVEC= 000014           210#
TST1    002452           750#
TST10   003202          1050#
TST100  012702          4155#
TST101  013022          4208#
TST102  013142          4261#
TST103  013262          4314#
TST104  013402          4367#
TST105  013522          4420#
TST106  013642          4474#
TST107  013760          4531#
TST11   003310          1105#
TST110  014074          4586#
TST111  014216          4640#
TST112  014336          4704#
TST113  014456          4763#
TST114  014636          4832#
TST115  015016          4894#
TST116  015136          4942#
TST117  015256          4994#
TST12   003416          1160#
TST120  015362          5052#
TST121  015600          5150#
TST122  015754          5218#
TST123  016104          5285#
TST124  016320          5369#
TST125  016510          5443#
TST126  016612          5481#
TST127  016720          5535#
TST13   003524          1215#
TST130  017046          5592#
TST131  017174          5649#
TST132  017322          5706#
TST133  017450          5763#
TST134  017576          5820#
TST135  017724          5878#
TST136  020116          5968#
TST137  020230          6030#
TST14   003632          1270#
TST140  020526          6141#
TST141  021024          6252#
TST142  021322          6363#
TST143  021620          6475#
TST144  022132          6589#
TST145  022444          6703#
TST146  022756          6817#
TST15   003740          1325#
TST16   004046          1380#
TST17   004154          1435#
TST2    002564           796#
```

```
TST20    004262            1490#
TST21    004370            1545#
TST22    004476            1600#
TST23    004604            1655#
TST24    004712            1710#
TST25    005020            1765#
TST26    005126            1820#
TST27    005234            1875#
TST3     002632             836#
TST30    005342            1930#
TST31    005450            1985#
TST32    005556            2040#
TST33    005664            2095#
TST34    005772            2150#
TST35    006100            2205#
TST36    006206            2260#
TST37    006314            2315#
TST4     002676             881#
TST40    006422            2370#
TST41    006530            2425#
TST42    006636            2480#
TST43    006744            2537#
TST44    007052            2593#
TST45    007160            2649#
TST46    007266            2705#
TST47    007374            2761#
TST5     002760             920#
TST50    007502            2817#
TST51    007610            2873#
TST52    007716            2929#
TST53    010024            2985#
TST54    010132            3040#
TST55    010240            3095#
TST56    010346            3150#
TST57    010454            3205#
TST6     003026             957#
TST60    010562            3260#
TST61    010670            3315#
TST62    010776            3370#
TST63    011104            3425#
TST64    011212            3486#
TST65    011272            3531#
TST66    011356            3581#
TST67    011442            3632#
TST7     003074             995#
TST70    011522            3678#
TST71    011606            3728#
TST72    011672            3776#
TST73    011756            3819#
TST74    012062            3869#
TST75    012204            3933#
TST76    012366            4011#
TST77    012534            4079#
TYPDS  = 104405            8211#
```

```
TYPE   = 104401          676   6941   7043   7050   7145   7152   7242   7249   7344   7351   7448   7455   7470
                        7478   7575   7613   7621   7650   7667   7669   7672   7674   7678   7685   7747   7879
                        7880   7883   7896   7907   7926   7975   7978   7982   8040   8167   8207#  8443   8497
                        8501   8505   8733   8919   8926   8943
TYPOC  = 104402         7476   7488   7658   7682   7882   8208#  8938
TYPON  = 104404         8210#
TYPOS  = 104403         8209#
VECTOR   001660          620#
VECTPS   001662          621#
VERSN    001664          623#
W1       033146         8631*  8646#
W2       033150         8635*  8647#
W3       033152         8640*  8648#
$AERR    032676         8450*  8543*  8569#  8859
$APTHD   001000          252    258#
$ASTAT=  ****** U       8110   8125
$ATYC    027426         8081   8083#
$ATY1    027402         8079#
$ATY3    027410         8025   8080#
$ATY4    027420         7626   8082#
$AUTOB   001134          290#   7876   7997
$BASE    001256          358#    681
$BDADR   001122          285#   8391
$BDDAT   001126          287#    759    761    763    765    767    769    807    846    891    930    967   1006
                        1026   1061   1081   1116   1136   1171   1191   1226   1246   1281   1301   1336   1356
                        1391   1411   1446   1466   1501   1521   1556   1576   1611   1631   1666   1686   1721
                        1741   1776   1796   1831   1851   1886   1906   1941   1961   1996   2016   2051   2071
                        2106   2126   2161   2181   2216   2236   2271   2291   2326   2346   2381   2401   2436
                        2456   2491   2511   2548   2568   2604   2624   2660   2680   2716   2736   2772   2792
                        2828   2848   2884   2904   2940   2960   2996   3016   3051   3071   3106   3126   3161
                        3181   3216   3236   3271   3291   3326   3346   3381   3401   3436   3456   3503   3550
                        3551   3600   3601   3649   3697   3698   3747   3748   3792   3834   3894   3968   3969
                        4034   4051   4123   4175   4181   4228   4234   4281   4287   4334   4340   4387   4393
                        4440   4446   4495   4500   4540   4559   4611   4666   4709   4730   4768   4796   4837
                        4865   4917   4965   5015   5095   5116   5175   5179   5244   5329   5330   5401   5417
                        5464   5507   5560   5565   5617   5622   5674   5679   5731   5736   5788   5793   5845
                        5850   5921   5993   6067   6081   6082   6109   6178   6192   6193   6220   6289   6303
                        6304   6331   6400   6414   6415   6442   6514   6529   6530   6558   6628   6643   6644
                        6672   6742   6757   6758   6786   6856   6871   6872   6900   7123   7221   7322   7426
                        8379   8381   8383   8385   8387   8389   8391   8397   8401
$BELL    001172          309#   7613   7642
$CDW1    001262          360#
$CDW2    001264          361#
$CHARC   027376         8042*  8052*  8059   8068*  8073#
$CKSWR   026344         7868#  8215
$CMTAG   001100          273#    632    633    641    647    648
$CM1   = 000001          305#    306#
$CM2   = 000002          305#    306#
$CM3   = 000001          303#    305
$CM4   = 000001          306#    307#
$CNT     033646         8837*  8839*  8845#
$CNTLG   027071         7879   7992#
$CNTLU   027064         7896   7991#
$CPUOP   001230          332#
```

```
$CRLF   001177   311#  7621  7642  7650  7669  7674  7678  7907  7991  8041  8076
$DATR   033320  8683*  8700* 8704# 8815
$DBLK   025752  7713   7747  7755#
$DDW0   001266   362#
$DDW1   001270   363#
$DDW10  001312   372#
$DDW11  001314   373#
$DDW12  001316   374#
$DDW13  001320   375#
$DDW14  001322   376#
$DDW15  001324   377#
$DDW2   001272   364#
$DDW3   001274   365#
$DDW4   001276   366#
$DDW5   001300   367#
$DDW6   001302   368#
$DDW7   001304   369#
$DDW8   001306   370#
$DDW9   001310   371#
$DEVcT  001212   323#  682*
$DEVM   001260   359#
$DOAGN  023354  6937   6943  6949#
$DTBL   025742  7716   7751#
$ENDAD  023344   238   6945#
$ENdCT  023324  6939#
$ENDMG  023363  6941   6953#
$ENuLL  023360  6952#
$ENV    001222   328#  7623  8020  8088  8112
$ENVM   001223   329#   669  8022  8027  8090
$EOP    023270  6929#
$EOPCT  023316  6936#  6940
$ERFLG  001103   276#  7608* 7642  7761  7790  7792  7798* 7820
$ERMAX  001115   282#   649* 7792  7815* 7820
$ERROR  025214   641   7606#
$ERRPC  001116   283#  7615* 7616* 7617  7642  7656  8379  8381  8383  8385  8387  8389  8391
                8393   8394  8395  8397  8399  8401  8403
$ERRTB  001360   416#  7664
$ERRTY  025402  7620   7649#
$ERTTL  001112   280#  7614* 7642
$ESCAP  001170   308#   648* 7637  7639  7642  7814*
$ETABL  001222   327#
$ETEND  001326   264    380#
$FATAL  001204   320#  8116#
$FFLG   027646  8079#  8082* 8110  8119* 8127#
$FILLC  001156   301#  8045  8076
$FILLS  001155   300#  8076
$GDADR  001120   284#  8391
$GDDAT  001124   286#   802*  805   839*  844   884*  889   905*  908   923*  928   960*  965
                1001*  1004  1006  1020* 1024  1056* 1059  1061  1075* 1079  1111* 1114  1116
                1130*  1134  1166* 1169  1171  1185* 1189  1221* 1224  1226  1240* 1244  1276*
                1279   1281  1295* 1299  1331* 1334  1336  1350* 1354  1386* 1389  1391  1405*
                1409   1441* 1444  1446  1460* 1464  1496* 1499  1501  1515* 1519  1551* 1554
                1556   1570* 1574  1606* 1609  1611  1625* 1629  1661* 1664  1666  1680* 1684
                1716*  1719  1721  1735* 1739  1771* 1774  1776  1790* 1794  1826* 1829  1831
```

```
                    1845*  1849   1881*  1884   1886   1900*  1904   1936*  1939   1941   1955*  1959   1991*
                    1994   1996   2010*  2014   2046*  2049   2051   2065*  2069   2101*  2104   2106   2120*
                    2124   2156*  2159   2161   2175*  2179   2211*  2214   2216   2230*  2234   2266*  2269
                    2271   2285*  2289   2321*  2324   2326   2340*  2344   2376*  2379   2381   2395*  2399
                    2431*  2434   2436   2450*  2454   2486*  2489   2491   2505*  2509   2543*  2546   2548
                    2562*  2566   2599*  2602   2604   2618*  2622   2655*  2658   2660   2674*  2678   2711*
                    2714   2716   2730*  2734   2767*  2770   2772   2786*  2790   2823*  2826   2828   2842*
                    2846   2879*  2882   2884   2898*  2902   2935*  2938   2940   2954*  2958   2991*  2994
                    2996   3010*  3014   3046*  3049   3051   3065*  3069   3101*  3104   3106   3120*  3124
                    3156*  3159   3161   3175*  3179   3211*  3214   3216   3230*  3234   3266*  3269   3271
                    3285*  3289   3321*  3324   3326   3340*  3344   3376*  3379   3381   3395*  3399   3431*
                    3434   3436   3450*  3454   3493*  3496   3498   3499*  3539*  3542   3543*  3548   3551
                    3585*  3588   3593*  3598   3601   3639*  3642   3644   3645*  3682*  3685   3690*  3693
                    3698   3732*  3735   3740*  3743   3748   3778*  3781   3783   3785*  3788   3824*  3827
                    3829*  3832   3850*  3853   3873*  3876   3878   3882*  3885   3887*  3890*  3894   3936*
                    3946   3963*  3969   3984   3989*  4014*  4017*  4020   4022*  4025   4027*  4030   4082*
                    4085   4091*  4094*  4097   4099*  4102   4161*  4164   4166*  4169   4214*  4217   4219*
                    4222   4267*  4270   4272*  4275   4320*  4323   4325*  4328   4373*  4376   4378*  4381
                    4426*  4429   4431*  4434   4480*  4483   4485*  4488   4541*  4544   4546*  4549   4551*
                    4554   4595*  4598*  4601*  4604   4650*  4653*  4656*  4659   4661*  4664   4711*  4714
                    4719*  4722   4724*  4727   4774*  4777*  4780*  4783*  4786   4788*  4791*  4796   4843*
                    4846*  4849*  4852*  4855   4857*  4860*  4865   4901*  4904*  4907*  4910   4912*  4915
                    4949*  4952*  4955*  4958   4960*  4963   5002*  5005*  5008   5010*  5013   5069*  5072
                    5074*  5077*  5080*  5083   5085*  5088   5091*  5134*  5137   5157*  5160*  5163*  5166*
                    5169   5192*  5195   5221*  5224   5226*  5229   5234*  5237   5239*  5242   5288*  5306
                    5324*  5330   5345   5350*  5380*  5383   5385*  5388*  5391*  5394   5396*  5399   5449*
                    5452   5454*  5457   5460*  5489*  5492   5494   5496*  5499   5539*  5545   5547   5549*
                    5552   5596*  5602   5604   5606*  5609   5653*  5659   5661   5663*  5666   5710*  5716
                    5718   5720*  5723   5767*  5773   5775   5777*  5780   5824*  5830   5832   5834*  5837
                    5895*  5898   5900   5902*  5905*  5908*  5911*  5914   5916*  5919   5980*  5983   5985*
                    5988   5989*  5993   6076*  6082   6109   6187*  6193   6220   6298*  6304   6331   6409*
                    6415   6442   6524*  6530   6558   6638*  6644   6672   6752*  6758   6786   6866*  6872
                    6900   8379   8381   8383   8385   8387   8389   8391   8395   8399
$GET42  023334      6942#
$GTSWR  026414      7880#  8213
$HD   = 000001        78     79
$HIBTS  001000       259#
$HIOCT  026342      7852*  7857#
$ICNT   001104       277#  6988*  7039*  7085*  7141*  7188*  7238*  7281*  7340*  7383*  7444*  7805*  7806
                    7808*  7819
$ILLUP  030010      8137   8153   8170#
$INLP   033514       759    761    763    765    767    769    807    846    891    930    967   1006   1026
                    1061   1081   1116   1136   1171   1191   1226   1246   1281   1301   1336   1356   1391
                    1411   1446   1466   1501   1521   1556   1576   1611   1631   1666   1686   1721   1741
                    1776   1796   1831   1851   1886   1906   1941   1961   1996   2016   2051   2071   2106
                    2126   2161   2181   2216   2236   2271   2291   2326   2346   2381   2401   2436   2456
                    2491   2511   2548   2568   2604   2624   2660   2680   2716   2736   2772   2792   2828
                    2848   2884   2904   2940   2960   2996   3016   3051   3071   3106   3126   3161   3181
                    3216   3236   3271   3291   3326   3346   3381   3401   3436   3456   3503   3550   3600
                    3649   3697   3747   3783   3792   3829   3834   3887   3894   3959   3968   4027   4034
                    4051   4099   4106   4123   4175   4181   4228   4234   4281   4287   4334   4340   4387
                    4393   4440   4446   4495   4500   4540   4559   4606   4611   4661   4666   4709   4724
                    4730   4768   4788   4796   4837   4857   4865   4912   4917   4960   4965   5010   5015
                    5085   5095   5116   5175   5179   5234   5239   5244   5321   5329   5396   5401   5417
```

```
                    5464    5507    5560    5565    5617    5622    5674    5679    5731    5736    5788    5793    5845
                    5850    5916    5921    5985    5993    6051    6062    6067    6081    6100    6109    6162    6173
                    6178    6192    6211    6220    6273    6284    6289    6303    6322    6331    6384    6395    6400
                    6414    6433    6442    6497    6509    6514    6529    6550    6558    6611    6623    6628    6643
                    6664    6672    6725    6737    6742    6757    6778    6786    6839    6851    6856    6871    6892
                    6900    7013    7015    7117    7123    7206    7221    7308    7322    7414    7426    8796#   8859
                    8933    8985    8992
$INTAG  001135       291#   7908    7997
$ITEMB  001114       281#   7617*   7625    7642    7653
$LF     001200       312#   7642    7982    7991    8076
$LFLG   027645      8120*   8126#
$LOAD   032700      8451    8579#
$LPADR  001106       278#    650*    752*   7796*   7812*   7817    7819
$LPAI   032020      8425#   8777    8822
$LPERR  001110       279#    651*   7636    7796    7813*   7819
$LPW    033322      8630    8634    8639    8643    8662    8665    8682    8699    8715#
$MADR1  001234       345#
$MADR2  001240       349#
$MADR3  001244       352#
$MADR4  001250       355#
$MAIL   001202       260     264     318#    668     7623    7811    8020
$MAMS1  001232       339#
$MAMS2  001236       347#
$MAMS3  001242       350#
$MAMS4  001246       353#
$MBADR  001002       260#
$MFLG   027644      8080*   8086    8121*   8125#
$MNeW   027107      7883    7995#
$MSGAD  001216       325#   8096*   8099
$MSGLG  001220       326#   8101*
$MSGTY  001202       319#   8094    8102*   8114    8118*
$MSwR   027076      7880    7993#
$MTYP1  001233       340#
$MTYP2  001237       348#
$MTYP3  001243       351#
$MTYP4  001247       354#
$MXCNT  026240      7809    7819#
$NULL   001154       299#   8047    8076
$NWTST= 000001       738#    740     771#    773     823#    825     862#    864     910#    912     948#    950     985#
                     987    1040#   1042    1095#   1097    1150#   1152    1205#   1207    1260#   1262    1315#   1317
                    1370#   1372    1425#   1427    1480#   1482    1535#   1537    1590#   1592    1645#   1647    1700#
                    1702    1755#   1757    1810#   1812    1865#   1867    1920#   1922    1975#   1977    2030#   2032
                    2085#   2087    2140#   2142    2195#   2197    2250#   2252    2305#   2307    2360#   2362    2415#
                    2417    2470#   2472    2527#   2529    2583#   2585    2639#   2641    2695#   2697    2751#   2753
                    2807#   2809    2863#   2865    2919#   2921    2975#   2977    3030#   3032    3085#   3087    3140#
                    3142    3195#   3197    3250#   3252    3305#   3307    3360#   3362    3415#   3417    3474#   3476
                    3519#   3521    3569#   3571    3620#   3622    3666#   3668    3716#   3718    3767#   3769    3809#
                    3811    3855#   3857    3914#   3916    3999#   4001    4067#   4069    4145#   4147    4198#   4200
                    4251#   4253    4304#   4306    4357#   4359    4410#   4412    4463#   4465    4519#   4521    4575#
                    4577    4626#   4628    4690#   4692    4744#   4746    4813#   4815    4884#   4886    4932#   4934
                    4981#   4983    5030#   5032    5138#   5140    5200#   5202    5269#   5271    5356#   5358    5433#
                    5435    5478#   5525#   5527    5582#   5584    5639#   5641    5696#   5698    5753#   5755    5810#
                    5812    5867#   5869    5949#   5951    6015#   6017    6126#   6128    6237#   6239    6348#   6350
                    6460#   6462    6574#   6576    6688#   6690    6802#   6804
```

```
$SETUP= 000117   629#  638   639   641   643   645   647   648   650   6931  7607  7633  7641
                7771  7863  7997
$STUP = 177777   629#
$SVLAD  026170  7781  7810#
$SVPC = 000234   236#  241
$SWR  = 167400    30#   78    83    84    85    86    87    88    89   307   308   309   647
                 648   650   651   751   797   837   882   921   958   996  1051  1106  1161
                1216  1271  1326  1381  1436  1491  1546  1601  1656  1711  1766  1821  1876
                1931  1986  2041  2096  2151  2206  2261  2316  2371  2426  2481  2538  2594
                2650  2706  2762  2818  2874  2930  2986  3041  3096  3151  3206  3261  3316
                3371  3426  3487  3532  3582  3633  3679  3729  3777  3820  3870  3934  4012
                4080  4156  4209  4262  4315  4368  4421  4475  4532  4587  4641  4705  4764
                4833  4895  4943  4995  5053  5151  5219  5286  5370  5444  5482  5536  5593
                5650  5707  5764  5821  5879  5969  6031  6142  6253  6364  6476  6590  6704
                6818  6924  6932  6944  6950  6952  7598  7599  7600  7601  7602  7611  7618
                7630  7634  7642  7762  7763  7764  7765  7766  7772  7784  7786  7787  7790
                7791  7792  7799  7800  7801  7813  7816  7819  8169
$SWREG  001224   330#   671
$SWRMK= 000000    89    90  7766  7767  7788
$TESTN  001206   321#   754*  7811*
$TIMES  001166   307#   647*   751*   797*   837*   882*   921*   958*   996*  1051*  1106*  1161*  1216*
                1271*  1326*  1381*  1436*  1491*  1546*  1601*  1656*  1711*  1766*  1821*  1876*  1931*
                1986*  2041*  2096*  2151*  2206*  2261*  2316*  2371*  2426*  2481*  2538*  2594*  2650*
                2706*  2762*  2818*  2874*  2930*  2986*  3041*  3096*  3151*  3206*  3261*  3316*  3371*
                3426*  3487*  3532*  3582*  3633*  3679*  3729*  3934*  4156*  4209*  4262*  4315*  4368*
                4421*  4475*  5151*  5286*  5444*  5482*  5536*  5593*  5650*  5707*  5764*  5821*  6031*
                6142*  6253*  6364*  6476*  6590*  6704*  6818*  6932*  7799*  7806  7809*  7819
$TKb    001146   296#  7861  7872  7889  7943  7949  8925
$TKS    001144   295#  7861  7870  7886  7910*  7941  7947  8923
$TLKR   033154  8659#  8811
$TLKW   033040  8627#  8769
$TMDAT  001356   399#  3941*  3944  3951*  3954  3959*  3962  4106  4606*  4609  5292*  5295  5296*
                5299  5309*  5312*  5315  5321*  5324  5977*  5980  6039*  6042  6044  6046*  6049
                6051*  6054  6062*  6065  6100*  6103  6150*  6153  6155  6157*  6160  6162*  6165
                6173*  6176  6211*  6214  6261*  6264  6266  6268*  6271  6273*  6276  6284*  6287
                6322*  6325  6372*  6375  6377  6379*  6382  6384*  6387  6395*  6398  6433*  6436
                6485*  6488  6490  6492*  6495  6497*  6500*  6503  6509*  6512  6550*  6553  6599*
                6602  6604  6606*  6609  6611*  6614*  6617  6623*  6626  6664*  6667  6713*  6716
                6718  6720*  6723  6725*  6728*  6731  6737*  6740  6778*  6781  6827*  6830  6832
                6834*  6837  6839*  6842*  6845  6851*  6854  6892*  6895  6996*  6999  7001  7013
                7015*  7018  7021  7093*  7096  7098  7108*  7111*  7112*  7115  7117*  7120  7196*
                7199  7201  7203*  7206*  7209  7289*  7292  7294  7296*  7299  7301*  7304  7308
                7311  7391*  7394  7396  7398*  7401*  7404  7408*  7411  7414
$TMPO   001164   306#  8395  8397  8399  8401  8403
$TN   = 000147    78#   738   751#   771   797#   823   837#   862   882#   910   921#   948   958#
                 985   996#  1040  1051#  1095  1106#  1150  1161#  1205  1216#  1260  1271#  1315
                1326#  1370  1381#  1425  1436#  1480  1491#  1535  1546#  1590  1601#  1645  1656#
                1700  1711#  1755  1766#  1810  1821#  1865  1876#  1920  1931#  1975  1986#  2030
                2041#  2085  2096#  2140  2151#  2195  2206#  2250  2261#  2305  2316#  2360  2371#
                2415  2426#  2470  2481#  2527  2538#  2583  2594#  2639  2650#  2695  2706#  2751
                2762#  2807  2818#  2863  2874#  2919  2930#  2975  2986#  3030  3041#  3085  3096#
                3140  3151#  3195  3206#  3250  3261#  3305  3316#  3360  3371#  3415  3426#  3474
                3487#  3519  3532#  3569  3582#  3620  3633#  3666  3679#  3716  3729#  3767  3777#
                3809  3820#  3855  3870#  3914  3934#  3999  4012#  4067  4080#  4145  4156#  4198
```

```
                    4209#  4251   4262#  4304   4315#  4357   4368#  4410   4421#  4463   4475#  4519   4532#
                    4575   4587#  4626   4641#  4690   4705#  4744   4764#  4813   4833#  4884   4895#  4932
                    4943#  4981   4995#  5030   5053#  5138   5151#  5200   5219#  5269   5286#  5356   5370#
                    5433   5444#  5478   5482#  5525   5536#  5582   5593#  5639   5650#  5696   5707#  5753
                    5764#  5810   5821#  5867   5879#  5949   5969#  6015   6031#  6126   6142#  6237   6253#
                    6348   6364#  6460   6476#  6574   6590#  6688   6704#  6802   6818#
$TOUT   033610      8473   8527   8554   8668   8665   8834#
$TPB    001152       298#  8065*  8076
$TPFLG  001157       302#  8014   8076
$TPS    001150       297#  8063   8076
$TRAP   030026       643   8184#
$TRAP2  030050      8195#  8206
$TRP  = 000014      8199#  8208#  8209#  8210#  8211#  8212#  8213   8214#  8215   8216#  8217#  8218#  8219#
                    8220#
$TRPAD  030062      8189   8206#
$TSTM   001004       261#
$TSTNM  001102       275#   753*  6931*  7610   7642   7761   7788   7810*  7811   7816   7820
$TTYIN  027054      7968   7969   7986   7990#
$TYPBN= ****** U    8212
$TYPDS  025536      7701#  8211
$TYPE   027120      8014#  8107   8199   8207
$TYPEC  027332      7912   8044   8051   8058   8063#  8064
$TYPEX  027400      8069   8071   8074#
$TYPOC  025012      7545#  8208
$TYPON  025026      7544   7547#  8210
$TYPOS  024766      7540#  8209
$UNIT   001214       324#
$UNITM  001010       263#
$USWR   001226       331#
$UTK    033772      8916#
$VECT1  001252       356#
$VECT2  001254       357#
$XTSTR  025774      7775#
$$GET4= 000000      6944#
$OFILL  025211      7541*  7545*  7555   7590#
$40CAT= ****** U    7620   7772
.       = 034276      92#    96#    99#   236    237#   239#   241#   242#   248    249#   251#   253#   272#
                     313    626#   636    650    651    679#  6952   6955   7053#  7155#  7252#  7354#  7458#
                    7481#  7642   7688#  7755#  7819   7820   7861   7990#  7991   7997   8076   8128#  8149
                    8171   8377#  8812   8922#
.DVLS   001666       625#   698*  8757   8778   8799   8823   8917*
.$ASTA= ****** U    8080   8083
.$X   = 001000       248#   253
```

```
                5181    5187    5246    5262    5332    5339    5403    5409    5420    5426    5467    5473    5509    5516    5569
                5575    5626    5632    5683    5689    5740    5746    5797    5803    5854    5860    5924    5938    5996    6005
                6084    6090    6112    6119    6195    6201    6223    6230    6306    6312    6334    6341    6417    6423    6445
                6452    6532    6539    6561    6568    6646    6653    6675    6682    6760    6767    6789    6796    6874    6881
                6903    6910    7023    7030    7126    7132    7223    7229    7324    7331    7428    7435    7496    7505
ENDCOM           219#    815     821     853     859     897     904     937     944     974     981    1012    1018    1032    1038
                1067    1073    1087    1093    1122    1128    1142    1148    1177    1183    1197    1203    1232    1238    1252
                1258    1287    1293    1307    1313    1342    1348    1362    1368    1397    1403    1417    1423    1452    1458
                1472    1478    1507    1513    1527    1533    1562    1568    1582    1588    1617    1623    1637    1643    1672
                1678    1692    1698    1727    1733    1747    1753    1782    1788    1802    1808    1837    1843    1857    1863
                1892    1898    1912    1918    1947    1953    1967    1973    2002    2008    2022    2028    2057    2063    2077
                2083    2112    2118    2132    2138    2167    2173    2187    2193    2222    2228    2242    2248    2277    2283
                2297    2303    2332    2338    2352    2358    2387    2393    2407    2413    2442    2448    2462    2468    2497
                2503    2517    2523    2554    2560    2574    2580    2610    2616    2630    2636    2666    2672    2686    2692
                2722    2728    2742    2748    2778    2784    2798    2804    2834    2840    2854    2860    2890    2896    2910
                2916    2946    2952    2966    2972    3002    3008    3022    3028    3057    3063    3077    3083    3112    3118
                3132    3138    3167    3173    3187    3193    3222    3228    3242    3248    3277    3283    3297    3303    3332
                3338    3352    3358    3387    3393    3407    3413    3442    3448    3462    3468    3510    3517    3559    3567
                3609    3617    3656    3663    3706    3714    3756    3764    3799    3806    3841    3849    3901    3911    3976
                3983    4040    4046    4058    4066    4112    4118    4130    4141    4189    4195    4242    4248    4295    4301
                4348    4354    4401    4407    4454    4460    4508    4516    4565    4572    4617    4623    4672    4687    4736
                4741    4802    4810    4871    4879    4923    4930    4971    4978    5021    5027    5101    5111    5124    5133
                5185    5191    5250    5266    5336    5343    5407    5413    5424    5430    5471    5477    5513    5520    5573
                5579    5630    5636    5687    5693    5744    5750    5801    5807    5858    5864    5928    5942    6000    6009
                6088    6094    6116    6123    6199    6205    6227    6234    6310    6316    6338    6345    6421    6427    6449
                6456    6536    6543    6565    6572    6650    6657    6679    6686    6764    6771    6793    6800    6878    6885
                6907    6914    7027    7034    7130    7136    7227    7233    7328    7335    7432    7439    7465    7500    7509
ERROR            113#    815     853     897     937     974    1012    1032    1067    1087    1122    1142    1177    1197    1232
                1252    1287    1307    1342    1362    1397    1417    1452    1472    1507    1527    1562    1582    1617    1637
                1672    1692    1727    1747    1782    1802    1837    1857    1892    1912    1947    1967    2002    2022    2057
                2077    2112    2132    2167    2187    2222    2242    2277    2297    2332    2352    2387    2407    2442    2462
                2497    2517    2554    2574    2610    2630    2666    2686    2722    2742    2778    2798    2834    2854    2890
                2910    2946    2966    3002    3022    3057    3077    3112    3132    3167    3187    3222    3242    3277    3297
                3332    3352    3387    3407    3442    3462    3510    3559    3609    3656    3706    3756    3799    3841    3901
                3976    4040    4058    4112    4130    4189    4242    4295    4348    4401    4454    4508    4565    4617    4672
                4736    4802    4871    4923    4971    5021    5101    5124    5185    5250    5336    5407    5424    5471    5513
                5573    5630    5687    5744    5801    5858    5928    6001    6088    6116    6199    6227    6310    6338    6421
                6449    6536    6565    6650    6679    6764    6793    6878    6907    7028    7130    7227    7328    7432    8469
                8475    8488    8529    8556    8670    8687
ESCAPE           219#
GETPRI           219#
GETSWR           219#
MOVEI             20#    757     759     761     763     765     767     805     844     889     928     965    1004    1024    1059
                1079    1114    1134    1169    1189    1224    1244    1279    1299    1334    1354    1389    1409    1444    1464
                1499    1519    1554    1574    1609    1629    1664    1684    1719    1739    1774    1794    1829    1849    1884
                1904    1939    1959    1994    2014    2049    2069    2104    2124    2159    2179    2214    2234    2269    2289
                2324    2344    2379    2399    2434    2454    2489    2509    2546    2566    2602    2622    2658    2678    2714
                2734    2770    2790    2826    2846    2882    2902    2938    2958    2994    3014    3049    3069    3104    3124
                3159    3179    3214    3234    3269    3289    3324    3344    3379    3399    3434    3454    3501    3548    3598
                3647    3695    3745    3781    3790    3827    3832    3885    3892    3956    3966    4025    4032    4049    4097
                4104    4120    4173    4179    4226    4232    4279    4285    4332    4338    4385    4391    4438    4444    4493
                4498    4538    4557    4604    4609    4659    4664    4707    4722    4728    4766    4786    4794    4835    4855
                4863    4910    4915    4958    4963    5008    5013    5083    5093    5114    5173    5177    5232    5237    5242
                5319    5327    5394    5399    5415    5462    5505    5558    5563    5615    5620    5672    5677    5729    5734
```

```
          5786    5791    5843    5848    5914    5919    5983    5991    6049    6060    6065    6079    6098    6107    6160
          6171    6176    6190    6209    6218    6271    6282    6287    6301    6320    6329    6382    6393    6398    6412
          6431    6440    6495    6507    6512    6527    6547    6556    6609    6621    6626    6641    6661    6670    6723
          6735    6740    6755    6775    6784    6837    6849    6854    6869    6889    6898    7011    7013    7115    7121
          7204    7219    7306    7320    7411    7424    8857
MOVEM       19#     803     842     887     906     926     963    1002    1022    1057    1077    1112    1132    1167    1187
          1222    1242    1277    1297    1332    1352    1387    1407    1442    1462    1497    1517    1552    1572    1607
          1627    1662    1682    1717    1737    1772    1792    1827    1847    1882    1902    1937    1957    1992    2012
          2047    2067    2102    2122    2157    2177    2212    2232    2267    2287    2322    2342    2377    2397    2432
          2452    2487    2507    2544    2564    2600    2620    2656    2676    2712    2732    2768    2788    2824    2844
          2880    2900    2936    2956    2992    3012    3047    3067    3102    3122    3157    3177    3212    3232    3267
          3287    3322    3342    3377    3397    3432    3452    3494    3496    3540    3546    3586    3596    3640    3642
          3683    3691    3733    3741    3779    3786    3825    3830    3851    3874    3876    3883    3888    3942    3944
          3952    3960    4015    4018    4023    4028    4083    4092    4095    4100    4162    4164    4167    4215    4217
          4220    4268    4270    4273    4321    4323    4326    4374    4376    4379    4427    4429    4432    4481    4483
          4486    4542    4544    4547    4552    4596    4599    4602    4607    4651    4654    4657    4662    4712    4720
          4725    4775    4778    4781    4784    4789    4844    4847    4850    4853    4858    4902    4905    4908    4913
          4950    4953    4956    4961    5003    5006    5011    5070    5072    5075    5078    5081    5086    5135    5158
          5161    5164    5167    5193    5222    5227    5235    5240    5293    5297    5304    5310    5313    5322    5381
          5383    5386    5389    5392    5397    5450    5452    5455    5490    5492    5494    5497    5543    5545    5547
          5550    5600    5602    5604    5607    5657    5659    5661    5664    5714    5716    5718    5721    5771    5773
          5775    5778    5828    5830    5832    5835    5896    5898    5900    5903    5906    5909    5912    5917    5978
          5981    5986    6040    6042    6044    6047    6052    6063    6101    6151    6153    6155    6158    6163    6174
          6212    6262    6264    6266    6269    6274    6285    6323    6373    6375    6377    6380    6385    6396    6434
          6486    6488    6490    6493    6498    6501    6510    6551    6600    6602    6604    6607    6612    6615    6624
          6665    6714    6716    6718    6721    6726    6729    6738    6779    6828    6830    6832    6835    6840    6843
          6852    6893    6997    6999    7016    7019    7094    7096    7109    7113    7118    7197    7199    7207    7290
          7292    7297    7302    7309    7392    7394    7399    7402    7409
MSY         227#     744     790     830     875     914     952     990    1045    1100    1155    1210    1265    1320    1375
          1430    1485    1540    1595    1650    1705    1760    1815    1870    1925    1980    2035    2090    2145    2200
          2255    2310    2365    2420    2475    2532    2588    2644    2700    2756    2812    2868    2924    2980    3035
          3090    3145    3200    3255    3310    3365    3420    3480    3525    3575    3626    3672    3722    3771    3814
          3864    3928    4007    4074    4150    4203    4256    4309    4362    4415    4470    4527    4582    4634    4699
          4758    4827    4889    4937    4989    5048    5146    5213    5281    5364    5438    5530    5587    5644    5701
          5758    5815    5873    5964    6026    6137    6248    6359    6471    6585    6699    6813    6977    7075    7177
          7271    7373
MTAGS       225#     382
MULT        219#
NEWTST      219#     738     771     823     862     910     948     985    1040    1095    1150    1205    1260    1315    1370
          1425    1480    1535    1590    1645    1700    1755    1810    1865    1920    1975    2030    2085    2140    2195
          2250    2305    2360    2415    2470    2527    2583    2639    2695    2751    2807    2863    2919    2975    3030
          3085    3140    3195    3250    3305    3360    3415    3474    3519    3569    3620    3666    3716    3767    3809
          3855    3914    3999    4067    4145    4198    4251    4304    4357    4410    4463    4519    4575    4626    4690
          4744    4813    4884    4932    4981    5030    5138    5200    5269    5356    5433    5478    5525    5582    5639
          5696    5753    5810    5867    5949    6015    6126    6237    6348    6460    6574    6688    6802
POP         219#    7742    7853    8122    8123    8158    8159
POPSP2      226#
PUSH        219#    7701    7832    8083    8085    8106    8139    8145
REPORT      219#
RTM        4142#   4143    4196    4249    4302    4355    4408
SCOPE       114#     796     836     881     920     957     995    1050    1105    1160    1215    1270    1325    1380    1435
          1490    1545    1600    1655    1710    1765    1820    1875    1930    1985    2040    2095    2150    2205    2260
          2315    2370    2425    2480    2537    2593    2649    2705    2761    2817    2873    2929    2985    3040    3095
          3150    3205    3260    3315    3370    3425    3486    3531    3581    3632    3678    3728    3776    3819    3869
```

```
        3933    4011    4079    4155    4208    4261    4314    4367    4420    4474    4531    4586    4640    4704    4763
        4832    4894    4942    4994    5052    5150    5218    5285    5369    5443    5481    5535    5592    5649    5706
        5763    5820    5878    5968    6030    6141    6252    6363    6475    6589    6703    6817
SETPRI   219#
SETTRA  8199#   8208    8209    8210    8211    8213    8215    8216    8217    8218    8219
SETUP    219#    630
SKIP     219#
SLASH    219#
SPACE    219#
STARS    219#     234     245     247     254     268     313     316     738     749     771     795     823     835     862
          880     910     919     948     956     985     994    1040    1049    1095    1104    1150    1159    1205    1214
         1260    1269    1315    1324    1370    1379    1425    1434    1480    1489    1535    1544    1590    1599    1645
         1654    1700    1709    1755    1764    1810    1819    1865    1874    1920    1929    1975    1984    2030    2039
         2085    2094    2140    2149    2195    2204    2250    2259    2305    2314    2360    2369    2415    2424    2470
         2479    2527    2536    2583    2592    2639    2648    2695    2704    2751    2760    2807    2816    2863    2872
         2919    2928    2975    2984    3030    3039    3085    3094    3140    3149    3195    3204    3250    3259    3305
         3314    3360    3369    3415    3424    3474    3485    3519    3530    3569    3580    3620    3631    3666    3677
         3716    3727    3767    3775    3809    3818    3855    3868    3914    3932    3999    4010    4067    4078    4145
         4154    4198    4207    4251    4260    4304    4313    4357    4366    4410    4419    4463    4473    4519    4530
         4575    4585    4626    4639    4690    4703    4744    4762    4813    4831    4884    4893    4932    4941    4981
         4993    5030    5051    5138    5149    5200    5217    5269    5284    5356    5368    5433    5442    5478    5480
         5525    5534    5582    5591    5639    5648    5696    5705    5753    5762    5810    5819    5867    5877    5949
         5967    6015    6029    6126    6140    6237    6251    6348    6362    6460    6474    6574    6588    6688    6702
         6802    6816    6921    7517    7594    7644    7691    7758    7822    7860    7863    7931    7960    7999    8078
         8135    8151    8178
STM1    6959#   6987    7084    7187    7280    7382
STM2    6960#   7035    7137    7234    7336    7440
SWRSU    219#    652#
TOUT     628#   8472    8526    8553    8667    8684
TRMTRP  8199#
TYPBIN   219#
TYPDEC   219#
TYPNAM   219#
TYPNUM   219#
TYPOCS    30#    219#
TYPOCT   219#   7475    7487    7656    7680    7881    8937
TYPTXT   219#    676    7043    7050    7145    7152    7242    7249    7344    7351    7448    7455    7470    7478    8443
        8497    8501    8505    8732    8918    8943
UPDATE   231#
ZP23    3473#   3476    3521    3571
ZP24    3619#   3622    3668    3718
ZP25    3766#
ZP26    3766#   3769
ZP27    3808#   3811
ZP28    3854#   3857
ZP29    3913#   3916
ZP3E    4144#   4147    4197#   4200    4250#   4253    4303#   4306    4356#   4359    4409#   4412
ZP3EA   4462#   4465
ZP30    3998#   4001
ZP31    4067#   4069
ZP310   4980#   4983
ZP311   5029#   5032
ZP312   5137#   5140
ZP33    4518#   4521
```

```
ZP34      4574#   4577
ZP35      4625#   4628
ZP36      4689#   4692
ZP37      4743#   4746    4812#   4815
ZP38      4882#
ZP39      4883#   4886    4931#   4934
ZP42      5199#   5202
ZP43      5268#   5271
ZP44      5355#   5358
ZP45      5432#   5435
ZP46E     5523#   5527    5580#   5584    5637#   5641    5694#   5698    5751#   5755    5808#   5812
ZP47      5866#   5869
ZP6E      5947#   5951
ZP61E     6013#   6017    6124#   6128    6235#   6239    6346#   6350    6462    6576    6690    6804
zZ1        983#    987    1042    1097    1152    1207    1262    1317    1372    1427    1482    1537    1592    1647    1702
          1757    1812    1867    1922    1977    2032    2087    2142    2197    2252    2307    2362    2417    2472    2529
          2585    2641    2697    2753    2809    2865    2921    2977    3032    3087    3142    3197    3252    3307    3362
          3417    3477    3522    3572    3623    3669    3719
Z1         737#    740
Z2         770#    773
Z3         822#    825
Z4         861#    864
Z6         909#    912
Z7         947#    950
$CAL.       30#   8408
$DMAST      30#
$DMDT       30#
$MMAST      30#
$$CMRE     266#    305
$$CMTM     266#    306
$$ESCA     219#
$$NEWT     219#    738     771     823     862     910     948     985    1040    1095    1150    1205    1260    1315    1370
          1425    1480    1535    1590    1645    1700    1755    1810    1865    1920    1975    2030    2085    2140    2195
          2250    2305    2360    2415    2470    2527    2583    2639    2695    2751    2807    2863    2919    2975    3030
          3085    3140    3195    3250    3305    3360    3415    3474    3519    3569    3620    3666    3716    3767    3809
          3855    3914    3999    4067    4145    4198    4251    4304    4357    4410    4463    4519    4575    4626    4690
          4744    4813    4884    4932    4981    5030    5138    5200    5269    5356    5433    5478    5525    5582    5639
          5696    5753    5810    5867    5949    6015    6126    6237    6348    6460    6574    6688    6802
$$SET      8199#   8208    8209    8210    8211    8213    8215    8216    8217    8218    8219
$$SETM      668#
$$SKIP      219#
.EQUAT       30#    109
.HEADE       30#     68
.KMADR       17#    594
.KSIS        22#    685
.LOADL       25#   8571
.LPAIN       22#   8409
.PUTCS       24#   8964
.RESET       22#   8846
.SETTR       30#
.SETUP       30#    629
.SWRHI       30#     79
.SWRLO       90#
.TRMTR       30#
```

```
        8676  8693  8721  8727  8734  8739  8764  8779  8807  8824  8838  8866  8920  8939  8944
        8953
CLR      634   647   648   668   683   698   703   839   905  1020  1075  1130  1185  1240  1295
        1350  1405  1460  1515  1570  1625  1680  1735  1790  1845  1900  1955  2010  2065  2120
        2175  2230  2285  2340  2395  2450  2505  2562  2618  2674  2730  2786  2842  2898  2954
        3010  3065  3120  3175  3230  3285  3340  3395  3450  3493  3499  3539  3585  3639  3645
        3682  3732  3778  3850  3873  3936  3941  4014  4082  4161  4169  4214  4222  4267  4275
        4320  4328  4373  4381  4426  4434  4480  4489  4541  4595  4650  4711  4774  4783  4843
        4852  4901  4949  5069  5134  5157  5163  5169  5192  5226  5288  5292  5380  5449  5460
        5489  5499  5539  5553  5596  5610  5653  5667  5710  5724  5767  5781  5824  5838  5895
        5980  6039  6150  6261  6372  6485  6599  6713  6827  6931  6932  6988  6996  7085  7093
        7111  7188  7196  7203  7281  7289  7383  7391  7557  7652  7712  7715  7799  7814  7837
        7838  7884  7885  8155  8450  8459  8545  8582  8583  8587  8591  8599  8716  8758  8776
        8800  8821  8837  8890  8893  8917
CLRB     7741  7798  7981  8042  8068  8119  8120  8121
CMP       635   658   696  1006  1061  1116  1171  1226  1281  1336  1391  1446  1501  1556  1611
        1666  1721  1776  1831  1886  1941  1996  2051  2106  2161  2216  2271  2326  2381  2436
        2491  2548  2604  2660  2716  2772  2828  2884  2940  2996  3051  3106  3161  3216  3271
        3326  3381  3436  3551  3601  3698  3748  3894  3969  4123  4796  4865  5095  5116  5417
        5993  6082  6109  6193  6220  6304  6331  6415  6442  6530  6558  6644  6672  6758  6786
        6872  6900  7490  7492  7733  7782  7806  7868  7874  7894  7901  7913  7915  7945  7951
        7954  7956  7969  8448  8512  8589  8595  8597  8761  8804  8834
CMPB     5330  7623  7788  7792  7876  7908  7973  7979  8020  8035  8037  8045  8066  8070  8088
        8484  8486  8491  8538  8565  8717  8930
DEC      6103  6214  6325  6436  6935  7660  8902
DECB     7565  7576  8049  8052
EMT       113
HALT       96  7500  7632  8016  8148  8170  8611  8738
INC       694   704  3963  4170  4223  4276  4329  4382  4435  4485  4490  4546  5312  5324  5391
        5496  5554  5611  5668  5725  5782  5839  5908  6073  6184  6295  6406  6500  6521  6553
        6614  6635  6667  6728  6749  6781  6842  6863  6895  6933  7039  7048  7141  7150  7238
        7247  7340  7349  7444  7453  7571  7579  7614  7719  7805  7923  8118  8156  8442  8460
        8493  8495  8543  8588  8600  8719  8725  8763  8806  8839
INCB     5170  5500  7608  7810  8072  8463  8514  8609  8896
IOT       114
JMP       100   102   103   104   105   106   107  6950
JSR       759   761   763   765   767   769   805   807   844   846   889   891   908   928   930
         965   967  1004  1006  1024  1026  1059  1061  1079  1081  1114  1116  1134  1136  1169
        1171  1189  1191  1224  1226  1244  1246  1279  1281  1299  1301  1334  1336  1354  1356
        1389  1391  1409  1411  1444  1446  1464  1466  1499  1501  1519  1521  1554  1556  1574
        1576  1609  1611  1629  1631  1664  1666  1684  1686  1719  1721  1739  1741  1774  1776
        1794  1796  1829  1831  1849  1851  1884  1886  1904  1906  1939  1941  1959  1961  1994
        1996  2014  2016  2049  2051  2069  2071  2104  2106  2124  2126  2159  2161  2179  2181
        2214  2216  2234  2236  2269  2271  2289  2291  2324  2326  2344  2346  2379  2381  2399
        2401  2434  2436  2454  2456  2489  2491  2509  2511  2546  2548  2566  2568  2602  2604
        2622  2624  2658  2660  2678  2680  2714  2716  2734  2736  2770  2772  2790  2792  2826
        2828  2846  2848  2882  2884  2902  2904  2938  2940  2958  2960  2994  2996  3014  3016
        3049  3051  3069  3071  3104  3106  3124  3126  3159  3161  3179  3181  3214  3216  3234
        3236  3269  3271  3289  3291  3324  3326  3344  3346  3379  3381  3399  3401  3434  3436
        3454  3456  3496  3498  3503  3542  3548  3550  3588  3598  3600  3642  3644  3649  3685
        3693  3697  3735  3743  3747  3781  3783  3788  3792  3827  3829  3832  3834  3853  3876
        3878  3885  3887  3890  3894  3944  3946  3954  3959  3962  3968  4017  4020  4025  4027
        4030  4034  4051  4085  4094  4097  4099  4102  4106  4123  4164  4166  4169  4175  4181
        4217  4219  4222  4228  4234  4270  4272  4275  4281  4287  4323  4325  4328  4334  4340
```

```
        4376  4378  4381  4387  4393  4429  4431  4434  4440  4446  4483  4485  4488  4495  4500
        4540  4544  4546  4549  4554  4559  4598  4601  4604  4606  4609  4611  4653  4656  4659
        4661  4664  4666  4709  4714  4722  4724  4727  4730  4768  4777  4780  4783  4786  4788
        4791  4796  4837  4846  4849  4852  4855  4857  4860  4865  4904  4907  4910  4912  4915
        4917  4952  4955  4958  4960  4963  4965  5005  5008  5010  5013  5015  5072  5074  5077
        5080  5083  5085  5088  5095  5116  5137  5160  5163  5166  5169  5175  5179  5195  5224
        5229  5234  5237  5239  5242  5244  5295  5299  5306  5312  5315  5321  5324  5329  5383
        5385  5388  5391  5394  5396  5399  5401  5417  5452  5454  5457  5458  5464  5492  5494
        5496  5499  5507  5545  5547  5549  5552  5560  5565  5602  5604  5606  5609  5617  5622
        5659  5661  5663  5666  5674  5679  5716  5718  5720  5723  5731  5736  5773  5775  5777
        5780  5788  5793  5830  5832  5834  5837  5845  5850  5898  5900  5902  5905  5908  5911
        5914  5916  5919  5921  5980  5983  5985  5988  5993  6042  6044  6046  6049  6051  6054
        6062  6065  6067  6081  6100  6103  6109  6153  6155  6157  6160  6162  6165  6173  6176
        6178  6192  6211  6214  6220  6264  6266  6268  6271  6273  6276  6284  6287  6289  6303
        6322  6325  6331  6375  6377  6379  6382  6384  6387  6395  6398  6400  6414  6433  6436
        6442  6488  6490  6492  6495  6497  6500  6503  6509  6512  6514  6529  6550  6553  6558
        6602  6604  6606  6609  6611  6614  6617  6623  6626  6628  6643  6664  6667  6672  6716
        6718  6720  6723  6725  6728  6731  6737  6740  6742  6757  6778  6781  6786  6830  6832
        6834  6837  6839  6842  6845  6851  6854  6856  6871  6892  6895  6900  6945  6999  7001
        7013  7015  7018  7021  7096  7098  7111  7115  7117  7120  7123  7199  7201  7206  7209
        7221  7292  7294  7299  7304  7308  7311  7322  7394  7396  7401  7404  7411  7414  7426
        7620  7626  7912  8025  8044  8051  8058  8107  8451  8473  8516  8518  8520  8527  8554
        8630  8634  8639  8643  8662  8665  8668  8682  8685  8699  8769  8777  8811  8822  8859
        8933  8950  8985  8992
MOV      633   637   639   640   641   642   643   644   645   646   650   651   654   655   656
         657   662   664   665   666   671   681   682   689   690   691   692   695   699   700
         706   708   710   712   714   716   718   720   722   724   726   727   728   729   751
         752   797   802   837   882   884   921   923   958   960   996  1001  1051  1056  1106
        1111  1161  1166  1216  1221  1271  1276  1326  1331  1381  1386  1436  1441  1491  1496
        1546  1551  1601  1606  1656  1661  1711  1716  1766  1771  1821  1826  1876  1881  1931
        1936  1986  1991  2041  2046  2096  2101  2151  2156  2206  2211  2261  2266  2316  2321
        2371  2376  2426  2431  2481  2485  2538  2543  2594  2599  2650  2655  2706  2711  2762
        2767  2818  2823  2874  2879  2930  2935  2986  2991  3041  3046  3096  3101  3151  3156
        3206  3211  3261  3266  3316  3321  3371  3376  3426  3431  3487  3532  3543  3582  3593
        3633  3679  3690  3729  3740  3785  3824  3882  3890  3934  3951  4017  4022  4091  4094
        4156  4166  4209  4219  4262  4272  4315  4325  4368  4378  4421  4431  4475  4551  4598
        4601  4653  4656  4719  4777  4780  4791  4846  4849  4860  4904  4907  4952  4955  5002
        5005  5074  5077  5080  5091  5151  5160  5166  5221  5286  5296  5309  5385  5388  5444
        5454  5482  5536  5549  5593  5606  5650  5663  5707  5720  5764  5777  5821  5834  5902
        5905  5911  5977  5989  6031  6046  6056  6076  6095  6142  6157  6167  6187  6206  6253
        6268  6278  6298  6317  6364  6379  6389  6409  6428  6476  6492  6503  6524  6545  6590
        6606  6617  6638  6659  6704  6720  6731  6752  6773  6818  6834  6845  6866  6887  6938
        6942  6989  7086  7108  7112  7189  7282  7296  7301  7384  7398  7401  7408  7468  7475
        7485  7487  7540  7548  7549  7550  7556  7563  7581  7582  7583  7584  7585  7610  7615
        7636  7639  7651  7656  7665  7670  7675  7677  7681  7702  7703  7704  7705  7706  7707
        7708  7713  7716  7736  7742  7743  7744  7745  7746  7748  7749  7777  7778  7780  7783
        7796  7808  7809  7812  7813  7816  7817  7830  7831  7832  7833  7834  7836  7851  7852
        7853  7854  7855  7881  7905  7910  7939  7940  7967  7968  7983  7984  7985  7986  8018
        8019  8024  8032  8047  8084  8085  8092  8096  8101  8102  8104  8106  8116  8122  8123
        8137  8138  8139  8140  8141  8142  8143  8144  8145  8146  8147  8153  8154  8158  8159
        8160  8161  8162  8163  8164  8165  8166  8184  8185  8189  8195  8196  8426  8441  8449
        8458  8462  8471  8511  8546  8579  8580  8581  8585  8592  8603  8604  8627  8628  8631
        8632  8635  8644  8659  8660  8666  8701  8715  8729  8754  8755  8757  8766  8768  8772
        8773  8775  8796  8797  8799  8809  8815  8816  8817  8820  8836  8888  8929  8937  8948
```

```
        8984  8990
MOVB     649   753   754  7541  7542  7545  7546  7547  7551  7554  7555  7574  7617  7625  7711
        7714  7728  7731  7740  7811  7815  7839  7872  7889  7943  7949  7972  7977  8029  8057
        8065  8079  8080  8082  8187  8510  8515  8517  8519  8524  8552  8633  8636  8638  8640
        8641  8642  8663  8664  8681  8683  8698  8700  8925  8988  8989  8995
NEG     7552  7710
NOP      750  6930  6946  6947  6948
RESET   6944  8856
ROL     7558  7560  7561  7562  7564  7842  7844  7846
RTI      663   730  7494  7586  7641  7750  7818  7856  7911  7959  7987  8034  8169  8197  8903
RTS     7679  8074  8124  8190  8547  8567  8605  8645  8702  8730  8774  8818  8842  8868  8895
        8898  8999
SUB     3989  5350  7469  7486  7616  7717  8099
TRAP    8199  8208  8209  8210  8211  8213  8215  8216  8217  8218  8219
TST      673   807   846   891   930   967  1026  1081  1136  1191  1246  1301  1356  1411  1466
        1521  1576  1631  1686  1741  1796  1851  1906  1961  2016  2071  2126  2181  2236  2291
        2346  2401  2456  2511  2568  2624  2680  2736  2792  2848  2904  2960  3016  3071  3126
        3181  3236  3291  3346  3401  3456  3503  3649  3792  3984  4175  4228  4281  4334  4387
        4440  4495  4540  4559  4666  4709  4768  4837  4917  4965  5015  5175  5244  5464  5507
        5560  5617  5674  5731  5788  5845  5921  6067  6178  6289  6400  6514  6628  6742  6856
        7123  7569  7630  7637  7683  7722  7732  7779  7803  7850  7903  7918  8031  8039  8061
        8094  8112  8114  8186  8608  8759  8767  8801  8810  8859  8886  8891
TSTB    5179  5345  5401  5565  5622  5679  5736  5793  5850  7221  7322  7426  7724  7738  7790
        7870  7886  7941  7947  8014  8063  8086  8097  8110  8540  8923
.ASCII   310   311  8337
.ASCIZ   309   312   679  6953  7046  7053  7148  7155  7245  7252  7347  7354  7451  7458  7473
        7481  7687  7991  7992  7993  7995  8173  8220  8225  8229  8233  8237  8241  8245  8249
        8253  8257  8263  8269  8273  8278  8283  8289  8295  8301  8307  8313  8319  8325  8331
        8346  8354  8357  8360  8366  8372  8446  8500  8504  8508  8736  8922  8946  8957
.ASECT    14
.BLKB   7990
.BLKW    626  7755
.BYTE    275   276   281   282   290   291   299   300   301   302   328   329   339   340   347
         348   350   351   353   354  6952  7587  7588  7589  7590  7627  7628  7988  7989  8125
        8126  8127
.DSABL  7928
.ENABL    30  7861
.END    9003
.ENDC     73    86    88    89    90    99   113   205   219   235   239   241   246   248   255
         269   273   275   303   306   307   308   309   310   314   317   339   347   350   353
         356   357   358   359   360   361   362   363   364   365   366   367   368   369   370
         371   372   373   374   375   376   377   378   382   402   623   629   637   638   641
         643   645   647   648   650   652   673   679   701   739   740   749   750   751   752
         753   772   773   795   796   797   798   812   815   818   821   824   825   835   836
         837   838   850   853   856   859   863   864   880   881   882   883   894   897   901
         904   911   912   919   920   921   922   934   937   941   944   949   950   956   957
         958   959   971   974   978   981   986   987   994   995   996   997  1009  1012  1015
        1018  1029  1032  1035  1038  1041  1042  1049  1050  1051  1052  1064  1067  1070  1073
        1084  1087  1090  1093  1096  1097  1104  1105  1106  1107  1119  1122  1125  1128  1139
        1142  1145  1148  1151  1152  1159  1160  1161  1162  1174  1177  1180  1183  1194  1197
        1200  1203  1206  1207  1214  1215  1216  1217  1229  1232  1235  1238  1249  1252  1255
        1258  1261  1262  1269  1270  1271  1272  1284  1287  1290  1293  1304  1307  1310  1313
        1316  1317  1324  1325  1326  1327  1339  1342  1345  1348  1359  1362  1365  1368  1371
        1372  1379  1380  1381  1382  1394  1397  1400  1403  1414  1417  1420  1423  1426  1427
```

```
                5185    5188    5191    5201    5202    5217    5218    5219    5247    5250    5263    5266    5270    5271    5284
                5285    5286    5287    5333    5336    5340    5343    5357    5358    5368    5369    5370    5404    5407    5410
                5413    5421    5424    5427    5430    5434    5435    5442    5443    5444    5445    5468    5471    5474    5477
                5479    5480    5481    5482    5483    5510    5513    5517    5520    5526    5527    5534    5535    5536    5537
                5570    5573    5576    5579    5583    5584    5591    5592    5593    5594    5627    5630    5633    5636    5640
                5641    5648    5649    5650    5651    5684    5687    5690    5693    5697    5698    5705    5706    5707    5708
                5741    5744    5747    5750    5754    5755    5762    5763    5764    5765    5798    5801    5804    5807    5811
                5812    5819    5820    5821    5822    5855    5858    5861    5864    5868    5869    5877    5878    5879    5925
                5928    5939    5942    5950    5951    5967    5968    5969    5997    6000    6006    6009    6016    6017    6029
                6030    6031    6032    6085    6088    6091    6094    6113    6116    6120    6123    6127    6128    6140    6141
                6142    6143    6196    6199    6202    6205    6224    6227    6231    6234    6238    6239    6251    6252    6253
                6254    6307    6310    6313    6316    6335    6338    6342    6345    6349    6350    6362    6363    6364    6365
                6418    6421    6424    6427    6446    6449    6453    6456    6461    6462    6474    6475    6476    6477    6533
                6536    6540    6543    6562    6565    6569    6572    6575    6576    6588    6589    6590    6591    6647    6650
                6654    6657    6676    6679    6683    6686    6689    6690    6702    6703    6704    6705    6761    6764    6768
                6771    6790    6793    6797    6800    6803    6804    6816    6817    6818    6819    6875    6878    6882    6885
                6904    6907    6911    6914    6922    6923    6924    6926    6928    6931    6937    6940    6941    6942    6944
                6950    6952    6955    7024    7027    7031    7034    7046    7053    7127    7130    7133    7136    7148    7155
                7224    7227    7230    7233    7245    7252    7325    7328    7332    7335    7347    7354    7429    7432    7436
                7439    7451    7458    7465    7473    7481    7497    7500    7506    7509    7518    7595    7598    7608    7615
                7620    7621    7622    7630    7641    7642    7645    7660    7689    7692    7759    7762    7767    7772    7774
                7785    7788    7789    7790    7792    7794    7801    7805    7810    7812    7816    7819    7820    7823    7825
                7858    7861    7862    7864    7892    7928    7932    7960    7961    7968    7970    7973    7975    7991    7997
                8000    8029    8079    8080    8083    8110    8125    8136    8145    8146    8152    8158    8159    8169    8176
                8179    8185    8188    8207    8208    8209    8210    8211    8212    8213    8214    8215    8216    8217    8218
                8219    8446    8476    8483    8500    8504    8508    8530    8537    8557    8564    8671    8678    8688    8695
                8736    8906    8922    8946
.EQUIV           113     114     122     167     168     169     170     171     172     173     174     175     176     195     196
                 197     198     199     200     201     202     203     204
.EVEN            317     679    7046    7053    7148    7155    7245    7252    7347    7354    7451    7458    7473    7481    7688
                8128    8175    8377    8446    8500    8504    8508    8736    8922    8946    8960
.GLOBL            14
.IDENT            30
.IF               69      86      87      88      89      90      99     111     177     205     234     237     239     245     247
                 254     268     272     274     303     306     307     308     309     313     314     316     339     347     350
                 353     356     357     358     359     360     361     362     363     364     365     366     367     368     369
                 370     371     372     373     374     375     376     377     378     382     620     629     632     637     639
                 641     643     645     647     648     650     668     678     701     738     740     749     751     752     753
                 771     773     795     797     798     812     815     818     821     823     825     835     837     838     850
                 853     856     859     862     864     880     882     883     894     897     901     904     910     912     919
                 921     922     934     937     941     944     948     950     956     958     959     971     974     978     981
                 985     987     994     996     997    1009    1012    1015    1018    1029    1032    1035    1038    1040    1042
                1049    1051    1052    1064    1067    1070    1073    1084    1087    1090    1093    1095    1097    1104    1106
                1107    1119    1122    1125    1128    1139    1142    1145    1148    1150    1152    1159    1161    1162    1174
                1177    1180    1183    1194    1197    1200    1203    1205    1207    1214    1216    1217    1229    1232    1235
                1238    1249    1252    1255    1258    1260    1262    1269    1271    1272    1284    1287    1290    1293    1304
                1307    1310    1313    1315    1317    1324    1326    1327    1339    1342    1345    1348    1359    1362    1365
                1368    1370    1372    1379    1381    1382    1394    1397    1400    1403    1414    1417    1420    1423    1425
                1427    1434    1436    1437    1449    1452    1455    1458    1469    1472    1475    1478    1480    1482    1489
                1491    1492    1504    1507    1510    1513    1524    1527    1530    1533    1535    1537    1544    1546    1547
                1559    1562    1565    1568    1579    1582    1585    1588    1590    1592    1599    1601    1602    1614    1617
                1620    1623    1634    1637    1640    1643    1645    1647    1654    1656    1657    1669    1672    1675    1678
                1689    1692    1695    1698    1700    1702    1709    1711    1712    1724    1727    1730    1733    1744    1747
                1750    1753    1755    1757    1764    1766    1767    1779    1782    1785    1788    1799    1802    1805    1808
```

1810	1812	1819	1821	1822	1834	1837	1840	1843	1854	1857	1860	1863	1865	1867
1874	1876	1877	1889	1892	1895	1898	1909	1912	1915	1918	1920	1922	1929	1931
1932	1944	1947	1950	1953	1964	1967	1970	1973	1975	1977	1984	1986	1987	1999
2002	2005	2008	2019	2022	2025	2028	2030	2032	2039	2041	2042	2054	2057	2060
2063	2074	2077	2080	2083	2085	2087	2094	2096	2097	2109	2112	2115	2118	2129
2132	2135	2138	2140	2142	2149	2151	2152	2164	2167	2170	2173	2184	2187	2190
2193	2195	2197	2204	2206	2207	2219	2222	2225	2228	2239	2242	2245	2248	2250
2252	2259	2261	2262	2274	2277	2280	2283	2294	2297	2300	2303	2305	2307	2314
2316	2317	2329	2332	2335	2338	2349	2352	2355	2358	2360	2362	2369	2371	2372
2384	2387	2390	2393	2404	2407	2410	2413	2415	2417	2424	2426	2427	2439	2442
2445	2448	2459	2462	2465	2468	2470	2472	2479	2481	2482	2494	2497	2500	2503
2514	2517	2520	2523	2527	2529	2536	2538	2539	2551	2554	2557	2560	2571	2574
2577	2580	2583	2585	2592	2594	2595	2607	2610	2613	2616	2627	2630	2633	2636
2639	2641	2648	2650	2651	2663	2666	2669	2672	2683	2686	2689	2692	2695	2697
2704	2706	2707	2719	2722	2725	2728	2739	2742	2745	2748	2751	2753	2760	2762
2763	2775	2778	2781	2784	2795	2798	2801	2804	2807	2809	2816	2818	2819	2831
2834	2837	2840	2851	2854	2857	2860	2863	2865	2872	2874	2875	2887	2890	2893
2896	2907	2910	2913	2916	2919	2921	2928	2930	2931	2943	2946	2949	2952	2963
2966	2969	2972	2975	2977	2984	2986	2987	2999	3002	3005	3008	3019	3022	3025
3028	3030	3032	3039	3041	3042	3054	3057	3060	3063	3074	3077	3080	3083	3085
3087	3094	3096	3097	3109	3112	3115	3118	3129	3132	3135	3138	3140	3142	3149
3151	3152	3164	3167	3170	3173	3184	3187	3190	3193	3195	3197	3204	3206	3207
3219	3222	3225	3228	3239	3242	3245	3248	3250	3252	3259	3261	3262	3274	3277
3280	3283	3294	3297	3300	3303	3305	3307	3314	3316	3317	3329	3332	3335	3338
3349	3352	3355	3358	3360	3362	3369	3371	3372	3384	3387	3390	3393	3404	3407
3410	3413	3415	3417	3424	3426	3427	3439	3442	3445	3448	3459	3462	3465	3468
3474	3476	3485	3487	3488	3507	3510	3514	3517	3519	3521	3530	3532	3533	3556
3559	3564	3567	3569	3571	3580	3582	3583	3606	3609	3614	3617	3620	3622	3631
3633	3634	3653	3656	3660	3663	3666	3668	3677	3679	3680	3703	3706	3711	3714
3716	3718	3727	3729	3730	3753	3756	3761	3764	3767	3769	3775	3777	3796	3799
3803	3806	3809	3811	3818	3820	3838	3841	3846	3849	3855	3857	3868	3870	3898
3901	3908	3911	3914	3916	3932	3934	3935	3973	3976	3980	3983	3999	4001	4010
4012	4037	4040	4043	4046	4055	4058	4063	4066	4067	4069	4078	4080	4109	4112
4115	4118	4127	4130	4138	4141	4145	4147	4154	4156	4157	4186	4189	4192	4195
4198	4200	4207	4209	4210	4239	4242	4245	4248	4251	4253	4260	4262	4263	4292
4295	4298	4301	4304	4306	4313	4315	4316	4345	4348	4351	4354	4357	4359	4366
4369	4369	4398	4401	4404	4407	4410	4412	4419	4421	4422	4451	4454	4457	4460
4463	4465	4473	4475	4476	4505	4508	4513	4516	4519	4521	4530	4532	4562	4565
4569	4572	4575	4577	4585	4587	4614	4617	4620	4623	4626	4628	4639	4641	4669
4672	4684	4687	4690	4692	4703	4705	4733	4736	4738	4741	4744	4746	4762	4764
4799	4802	4807	4810	4813	4815	4831	4833	4868	4871	4876	4879	4884	4886	4893
4895	4920	4923	4927	4930	4932	4934	4941	4943	4968	4971	4975	4978	4981	4983
4993	4995	5018	5021	5024	5027	5030	5032	5051	5053	5098	5101	5108	5111	5121
5124	5130	5133	5138	5140	5149	5151	5152	5182	5185	5188	5191	5200	5202	5217
5219	5247	5250	5263	5266	5269	5271	5284	5286	5287	5333	5336	5340	5343	5356
5358	5368	5370	5404	5407	5410	5413	5421	5424	5427	5430	5433	5435	5442	5444
5445	5468	5471	5474	5477	5478	5480	5482	5483	5510	5513	5517	5520	5525	5527
5534	5536	5537	5570	5573	5576	5579	5582	5584	5591	5593	5594	5627	5630	5633
5636	5639	5641	5648	5650	5651	5684	5687	5690	5693	5696	5698	5705	5707	5708
5741	5744	5747	5750	5753	5755	5762	5764	5765	5798	5801	5804	5807	5810	5812
5819	5821	5822	5855	5858	5861	5864	5867	5869	5877	5879	5925	5928	5939	5942
5949	5951	5967	5969	5997	6000	6006	6009	6015	6017	6029	6031	6032	6085	6088
6091	6094	6113	6116	6120	6123	6126	6128	6140	6142	6143	6196	6199	6202	6205
6224	6227	6231	6234	6237	6239	6251	6253	6254	6307	6310	6313	6316	6335	6338

```
                6342    6345    6348    6350    6362    6364    6365    6418    6421    6424    6427    6446    6449    6453    6456
                6460    6462    6474    6476    6477    6533    6536    6540    6543    6562    6565    6569    6572    6574    6576
                6588    6590    6591    6647    6650    6654    6657    6676    6679    6683    6686    6688    6690    6702    6704
                6705    6761    6764    6768    6771    6790    6793    6797    6800    6802    6804    6816    6818    6819    6875
                6878    6882    6885    6904    6907    6911    6914    6921    6922    6923    6924    6925    6926    6930    6936
                6939    6941    6942    6944    6950    6952    6953    6955    7024    7027    7031    7034    7045    7052    7127
                7130    7133    7136    7147    7154    7224    7227    7230    7233    7244    7251    7325    7328    7332    7335
                7346    7353    7429    7432    7436    7439    7450    7457    7465    7472    7480    7497    7500    7506    7509
                7517    7594    7597    7608    7611    7618    7620    7621    7623    7630    7634    7641    7642    7644    7659
                7675    7691    7758    7761    7766    7772    7784    7786    7787    7788    7790    7791    7792    7801    7803
                7811    7813    7818    7819    7820    7822    7825    7837    7860    7862    7863    7864    7892    7931    7932
                7960    7968    7969    7973    7974    7990    7991    7997    7999    8020    8078    8080    8083    8110    8125
                8135    8145    8146    8151    8158    8159    8167    8169    8173    8178    8184    8188    8199    8208    8209
                8210    8211    8212    8213    8215    8216    8217    8218    8219    8445    8475    8481    8499    8503    8507
                8529    8535    8556    8562    8670    8676    8687    8693    8735    8906    8921    8945
.IFF              86      88      89      90     111     235     239     241     246     248     255     269     272     275     303
                 314     317     637     738     739     740     750     751     772     773     796     797     824     825     836
                 837     863     864     881     882     911     912     920     921     949     950     957     958     986     987
                 995     996    1041    1042    1050    1051    1096    1097    1105    1106    1151    1152    1160    1161    1206
                1207    1215    1216    1261    1262    1270    1271    1316    1317    1325    1326    1371    1372    1380    1381
                1426    1427    1435    1436    1481    1482    1490    1491    1536    1537    1545    1546    1591    1592    1600
                1601    1646    1647    1655    1656    1701    1702    1710    1711    1756    1757    1765    1766    1811    1812
                1820    1821    1866    1867    1875    1876    1921    1922    1930    1931    1976    1977    1985    1986    2031
                2032    2040    2041    2086    2087    2095    2096    2141    2142    2150    2151    2196    2197    2205    2206
                2251    2252    2260    2261    2306    2307    2315    2316    2361    2362    2370    2371    2416    2417    2425
                2426    2471    2472    2480    2481    2528    2529    2537    2538    2584    2585    2593    2594    2640    2641
                2649    2650    2696    2697    2705    2706    2752    2753    2761    2762    2808    2809    2817    2818    2864
                2865    2873    2874    2920    2921    2929    2930    2976    2977    2985    2986    3031    3032    3040    3041
                3086    3087    3095    3096    3141    3142    3150    3151    3196    3197    3205    3206    3251    3252    3260
                3261    3306    3307    3315    3316    3361    3362    3370    3371    3416    3417    3425    3426    3475    3476
                3486    3487    3520    3521    3531    3532    3570    3571    3581    3582    3621    3622    3632    3633    3667
                3668    3678    3679    3717    3718    3728    3729    3768    3769    3776    3777    3810    3811    3819    3820
                3856    3857    3869    3870    3915    3916    3933    3934    3935    4000    4001    4011    4012    4068    4069
                4079    4080    4146    4147    4155    4156    4199    4200    4208    4209    4252    4253    4261    4262    4305
                4306    4314    4315    4358    4359    4367    4368    4411    4412    4420    4421    4464    4465    4474    4475
                4520    4521    4531    4532    4576    4577    4586    4587    4627    4628    4640    4641    4691    4692    4704
                4705    4745    4746    4763    4764    4814    4815    4832    4833    4885    4886    4894    4895    4933    4934
                4942    4943    4982    4983    4994    4995    5031    5032    5052    5053    5139    5140    5150    5151    5201
                5202    5218    5219    5270    5271    5285    5286    5287    5357    5358    5369    5370    5434    5435    5443
                5444    5445    5479    5480    5481    5482    5526    5527    5535    5536    5583    5584    5592    5593    5640
                5641    5649    5650    5697    5698    5706    5707    5754    5755    5763    5764    5811    5812    5820    5821
                5868    5869    5878    5879    5950    5951    5968    5969    6016    6017    6030    6031    6127    6128    6141
                6142    6238    6239    6252    6253    6349    6350    6363    6364    6461    6462    6475    6476    6575    6576
                6589    6590    6689    6690    6703    6704    6803    6804    6817    6818    6922    6925    6930    6937    6940
                6952    7518    7595    7597    7611    7641    7642    7645    7660    7689    7692    7759    7785    7788    7789
                7792    7819    7823    7861    7864    7932    7934    7939    7960    7961    7970    7974    7991    8000    8079
                8136    8152    8169    8179    8185    8475    8481    8529    8535    8556    8562    8670    8676    8687    8693
.IFT             679    7046    7053    7148    7155    7245    7252    7347    7354    7451    7458    7473    7481    7621    7800
                7841    7857    7858    7934    7939    8446    8500    8504    8508    8736    8922    8946
.IFTF            679    7046    7053    7148    7155    7245    7252    7347    7354    7451    7458    7473    7481    7620    7798
                7837    7841    7857    7879    7932    7935    8446    8500    8504    8508    8736    8922    8946
.IIF              68      73      78      79      83      84      85      86      89      90      96     313     317     638     641
                 647     648     650     651    6923    6924    6931    6932    6952    6955    7476    7488    7598    7599    7600
                7601    7602    7607    7633    7641    7642    7657    7682    7762    7763    7764    7765    7766    7767    7771
```

	7799	7800	7816	7819	7820	7861	7882	7983	7991	7997	8076	8207	8208	8209	8210
	8211	8213	8215	8216	8217	8218	8219	8938							
.IRP	382	629	738	771	823	862	910	948	985	1040	1095	1150	1205	1260	1315
	1370	1425	1480	1535	1590	1645	1700	1755	1810	1865	1920	1975	2030	2085	2140
	2195	2250	2305	2360	2415	2470	2527	2583	2639	2695	2751	2807	2863	2919	2975
	3030	3085	3140	3195	3250	3305	3360	3415	3474	3519	3569	3620	3666	3716	3767
	3809	3855	3914	3999	4067	4145	4198	4251	4304	4357	4410	4463	4519	4575	4626
	4690	4744	4813	4884	4932	4981	5030	5138	5200	5269	5356	5433	5478	5525	5582
	5639	5696	5753	5810	5867	5949	6015	6126	6237	6348	6460	6574	6688	6802	6930
	7702	7742	7832	7853	8084	8085	8106	8122	8123	8139	8145	8158	8159		
.LIST	14	30	89	96	219	303	305	306	307	314	317	629	652	679	738
	751	759	761	763	765	767	769	771	797	805	807	812	815	818	821
	823	837	844	846	850	853	856	859	862	882	889	891	894	897	901
	904	908	910	921	928	930	934	937	941	944	948	958	965	967	971
	974	978	981	984	985	996	997	1004	1006	1009	1012	1015	1018	1024	1026
	1029	1032	1035	1038	1039	1040	1051	1052	1059	1061	1064	1067	1070	1073	1079
	1081	1084	1087	1090	1093	1094	1095	1106	1107	1114	1116	1119	1122	1125	1128
	1134	1136	1139	1142	1145	1148	1149	1150	1161	1162	1169	1171	1174	1177	1180
	1183	1189	1191	1194	1197	1200	1203	1204	1205	1216	1217	1224	1226	1229	1232
	1235	1238	1244	1246	1249	1252	1255	1258	1259	1260	1271	1272	1279	1281	1284
	1287	1290	1293	1299	1301	1304	1307	1310	1313	1314	1315	1326	1327	1334	1336
	1339	1342	1345	1348	1354	1356	1359	1362	1365	1368	1369	1370	1381	1382	1389
	1391	1394	1397	1400	1403	1409	1411	1414	1417	1420	1423	1424	1425	1436	1437
	1444	1446	1449	1452	1455	1458	1464	1466	1469	1472	1475	1478	1479	1480	1491
	1492	1499	1501	1504	1507	1510	1513	1519	1521	1524	1527	1530	1533	1534	1535
	1546	1547	1554	1556	1559	1562	1565	1568	1574	1576	1579	1582	1585	1588	1589
	1590	1601	1602	1609	1611	1614	1617	1620	1623	1629	1631	1634	1637	1640	1643
	1644	1645	1656	1657	1664	1666	1669	1672	1675	1678	1684	1686	1689	1692	1695
	1698	1699	1700	1711	1712	1719	1721	1724	1727	1730	1733	1739	1741	1744	1747
	1750	1753	1754	1755	1766	1767	1774	1776	1779	1782	1785	1788	1794	1796	1799
	1802	1805	1808	1809	1810	1821	1822	1829	1831	1834	1837	1840	1843	1849	1851
	1854	1857	1860	1863	1864	1865	1876	1877	1884	1886	1889	1892	1895	1898	1904
	1906	1909	1912	1915	1918	1919	1920	1931	1932	1939	1941	1944	1947	1950	1953
	1959	1961	1964	1967	1970	1973	1974	1975	1986	1987	1994	1996	1999	2002	2005
	2008	2014	2016	2019	2022	2025	2028	2029	2030	2041	2042	2049	2051	2054	2057
	2060	2063	2069	2071	2074	2077	2080	2083	2084	2085	2096	2097	2104	2106	2109
	2112	2115	2118	2124	2126	2129	2132	2135	2138	2139	2140	2151	2152	2159	2161
	2164	2167	2170	2173	2179	2181	2184	2187	2190	2193	2194	2195	2206	2207	2214
	2216	2219	2222	2225	2228	2234	2236	2239	2242	2245	2248	2249	2250	2261	2262
	2269	2271	2274	2277	2280	2283	2289	2291	2294	2297	2300	2303	2304	2305	2316
	2317	2324	2326	2329	2332	2335	2338	2344	2346	2349	2352	2355	2358	2359	2360
	2371	2372	2379	2381	2384	2387	2390	2393	2399	2401	2404	2407	2410	2413	2414
	2415	2426	2427	2434	2436	2439	2442	2445	2448	2454	2456	2459	2462	2465	2468
	2469	2470	2481	2482	2489	2491	2494	2497	2500	2503	2509	2511	2514	2517	2520
	2523	2524	2526	2527	2538	2539	2546	2548	2551	2554	2557	2560	2566	2568	2571
	2574	2577	2580	2582	2583	2594	2595	2602	2604	2607	2610	2613	2616	2622	2624
	2627	2630	2633	2636	2638	2639	2650	2651	2658	2660	2663	2666	2669	2672	2678
	2680	2683	2686	2689	2692	2694	2695	2706	2707	2714	2716	2719	2722	2725	2728
	2734	2736	2739	2742	2745	2748	2750	2751	2762	2763	2770	2772	2775	2778	2781
	2784	2790	2792	2795	2798	2801	2804	2806	2807	2818	2819	2826	2828	2831	2834
	2837	2840	2846	2848	2851	2854	2857	2860	2862	2863	2874	2875	2882	2884	2887
	2890	2893	2896	2902	2904	2907	2910	2913	2916	2918	2919	2930	2931	2938	2940
	2943	2946	2949	2952	2958	2960	2963	2966	2969	2972	2974	2975	2986	2987	2994
	2996	2999	3002	3005	3008	3014	3016	3019	3022	3025	3028	3029	3030	3041	3042

3049	3051	3054	3057	3060	3063	3069	3071	3074	3077	3080	3083	3084	3085	3096
3097	3104	3106	3109	3112	3115	3118	3124	3126	3129	3132	3135	3138	3139	3140
3151	3152	3159	3161	3164	3167	3170	3173	3179	3181	3184	3187	3190	3193	3194
3195	3206	3207	3214	3216	3219	3222	3225	3228	3234	3236	3239	3242	3245	3248
3249	3250	3261	3262	3269	3271	3274	3277	3280	3283	3289	3291	3294	3297	3300
3303	3304	3305	3316	3317	3324	3326	3329	3332	3335	3338	3344	3346	3349	3352
3355	3358	3359	3360	3371	3372	3379	3381	3384	3387	3390	3393	3399	3401	3404
3407	3410	3413	3414	3415	3426	3427	3434	3436	3439	3442	3445	3448	3454	3456
3459	3462	3465	3468	3469	3474	3487	3496	3498	3503	3507	3510	3514	3517	3519
3532	3542	3548	3550	3556	3559	3564	3567	3569	3582	3588	3598	3600	3606	3609
3614	3617	3620	3633	3642	3644	3649	3653	3656	3660	3663	3666	3679	3685	3693
3697	3703	3706	3711	3714	3716	3729	3735	3743	3747	3753	3756	3761	3764	3767
3777	3781	3783	3788	3792	3796	3799	3803	3806	3809	3820	3827	3829	3832	3834
3838	3841	3846	3849	3853	3855	3870	3876	3878	3885	3887	3890	3894	3898	3901
3908	3911	3914	3934	3944	3946	3954	3959	3962	3968	3973	3976	3980	3983	3999
4012	4017	4020	4025	4027	4030	4034	4037	4040	4043	4046	4051	4055	4058	4063
4066	4067	4080	4085	4094	4097	4099	4102	4106	4109	4112	4115	4118	4123	4127
4130	4138	4141	4143	4145	4156	4164	4166	4169	4175	4181	4186	4189	4192	4195
4196	4198	4209	4217	4219	4222	4228	4234	4239	4242	4245	4248	4249	4251	4262
4270	4272	4275	4281	4287	4292	4295	4298	4301	4302	4304	4315	4323	4325	4328
4334	4340	4345	4348	4351	4354	4355	4357	4368	4376	4378	4381	4387	4393	4398
4401	4404	4407	4408	4410	4421	4429	4431	4434	4440	4446	4451	4454	4457	4460
4463	4475	4483	4485	4488	4495	4500	4505	4508	4513	4516	4519	4532	4540	4544
4546	4549	4554	4559	4562	4565	4569	4572	4575	4587	4598	4601	4604	4606	4609
4611	4614	4617	4620	4623	4626	4641	4653	4656	4659	4661	4664	4666	4669	4672
4684	4687	4690	4705	4709	4714	4722	4724	4727	4730	4733	4736	4738	4741	4744
4764	4768	4777	4780	4783	4786	4788	4791	4796	4799	4802	4807	4810	4813	4833
4837	4846	4849	4852	4855	4857	4860	4865	4868	4871	4876	4879	4884	4895	4904
4907	4910	4912	4915	4917	4920	4923	4927	4930	4932	4943	4952	4955	4958	4960
4963	4965	4968	4971	4975	4978	4981	4995	5005	5008	5010	5013	5015	5018	5021
5024	5027	5030	5053	5072	5074	5077	5080	5083	5085	5088	5095	5098	5101	5108
5111	5116	5121	5124	5130	5133	5137	5138	5151	5160	5163	5166	5169	5175	5179
5182	5185	5188	5191	5195	5200	5219	5224	5229	5234	5237	5239	5242	5244	5247
5250	5263	5266	5269	5286	5295	5299	5306	5312	5315	5321	5324	5329	5333	5336
5340	5343	5356	5370	5383	5385	5388	5391	5394	5396	5399	5401	5404	5407	5410
5413	5417	5421	5424	5427	5430	5433	5444	5452	5454	5457	5464	5468	5471	5474
5477	5478	5482	5492	5494	5496	5499	5507	5510	5513	5517	5520	5523	5525	5536
5545	5547	5549	5552	5560	5565	5570	5573	5576	5579	5580	5582	5593	5602	5604
5606	5609	5617	5622	5627	5630	5633	5636	5637	5639	5650	5659	5661	5663	5666
5674	5679	5684	5687	5690	5693	5694	5696	5707	5716	5718	5720	5723	5731	5736
5741	5744	5747	5750	5751	5753	5764	5773	5775	5777	5780	5788	5793	5798	5801
5804	5807	5808	5810	5821	5830	5832	5834	5837	5845	5850	5855	5858	5861	5864
5867	5879	5898	5900	5902	5905	5908	5911	5914	5916	5919	5921	5925	5928	5939
5942	5949	5969	5980	5983	5985	5988	5993	5997	6000	6006	6009	6013	6015	6031
6042	6044	6046	6049	6051	6054	6062	6065	6067	6081	6085	6088	6091	6094	6100
6103	6109	6113	6116	6120	6123	6124	6126	6142	6153	6155	6157	6160	6162	6165
6173	6176	6178	6192	6196	6199	6202	6205	6211	6214	6220	6224	6227	6231	6234
6235	6237	6253	6264	6266	6268	6271	6273	6276	6284	6287	6289	6303	6307	6310
6313	6316	6322	6325	6331	6335	6338	6342	6345	6346	6348	6364	6375	6377	6379
6382	6384	6387	6395	6398	6400	6414	6418	6421	6424	6427	6433	6436	6442	6446
6449	6453	6456	6460	6476	6488	6490	6492	6495	6497	6500	6503	6509	6512	6514
6529	6533	6536	6540	6543	6550	6553	6558	6562	6565	6569	6572	6574	6590	6602
6604	6606	6609	6611	6614	6617	6623	6626	6628	6643	6647	6650	6654	6657	6664
6667	6672	6676	6679	6683	6686	6688	6704	6716	6718	6720	6723	6725	6728	6731

```
                6737    6740    6742    6757    6761    6764    6768    6771    6778    6781    6786    6790    6793    6797    6800
                6802    6818    6830    6832    6834    6837    6839    6842    6845    6851    6854    6856    6871    6875    6878
                6882    6885    6892    6895    6900    6904    6907    6911    6914    6931    6944    6999    7001    7013    7015
                7018    7021    7024    7027    7031    7034    7046    7053    7096    7098    7111    7115    7117    7120    7123
                7127    7130    7133    7136    7148    7155    7199    7201    7206    7209    7221    7224    7227    7230    7233
                7245    7252    7292    7294    7299    7304    7308    7311    7322    7325    7328    7332    7335    7347    7354
                7394    7396    7401    7404    7411    7414    7426    7429    7432    7436    7439    7451    7458    7466    7473
                7481    7497    7500    7506    7509    7641    7766    7960    8199    8207    8208    8209    8210    8211    8212
                8213    8214    8215    8216    8217    8218    8219    8220    8446    8500    8504    8508    8736    8859    8922
                8946
.MACRO            17      18      19      20      22      24      25      26      27      28      29      30      90     225     226
                 227     229     231     266     417     628     668     736     737     770     822     861     909     947     983
                 984    2524    3473    3619    3766    3808    3854    3913    3998    4067    4142    4144    4197    4250    4303
                4356    4409    4462    4518    4574    4625    4689    4743    4812    4882    4883    4931    4980    5029    5137
                5199    5268    5355    5432    5522    5523    5580    5637    5694    5751    5808    5866    5947    6012    6013
                6124    6235    6346    6458    6916    6959    6960    8199
.MCALL            30     219     314     652
.NLIST            14      30      89      96     219     303     305     306     307     314     317     629     652     679     738
                 751     759     761     763     765     767     769     771     797     805     807     812     815     818     821
                 823     837     844     846     850     853     856     859     862     882     889     891     894     897     901
                 904     908     910     921     928     930     934     937     941     944     948     958     961     967     971
                 974     978     981     984     985     996     997    1004    1006    1009    1012    1015    1018    1024    1026
                1029    1032    1035    1038    1039    1040    1051    1052    1059    1061    1064    1067    1070    1073    1079
                1081    1084    1087    1090    1093    1094    1095    1106    1107    1114    1116    1119    1122    1125    1128
                1134    1136    1139    1142    1145    1148    1149    1150    1161    1162    1169    1171    1174    1177    1180
                1183    1189    1191    1194    1197    1200    1203    1204    1205    1216    1217    1224    1226    1229    1232
                1235    1238    1244    1246    1249    1252    1255    1258    1259    1260    1271    1272    1279    1281    1284
                1287    1290    1293    1299    1301    1304    1307    1310    1313    1314    1315    1326    1327    1334    1336
                1339    1342    1345    1348    1354    1356    1359    1362    1365    1368    1369    1370    1381    1382    1389
                1391    1394    1397    1400    1403    1409    1411    1414    1417    1420    1423    1424    1425    1436    1437
                1444    1446    1449    1452    1455    1458    1464    1466    1469    1472    1475    1478    1479    1480    1491
                1492    1499    1501    1504    1507    1510    1513    1519    1521    1524    1527    1530    1533    1534    1535
                1546    1547    1554    1556    1559    1562    1565    1568    1574    1576    1579    1582    1585    1588    1589
                1590    1601    1602    1609    1611    1614    1617    1620    1623    1629    1631    1634    1637    1640    1643
                1644    1645    1656    1657    1664    1666    1669    1672    1675    1678    1684    1686    1689    1692    1695
                1698    1699    1700    1711    1712    1719    1721    1724    1727    1730    1733    1739    1741    1744    1747
                1750    1753    1754    1755    1766    1767    1774    1776    1779    1782    1785    1788    1794    1796    1799
                1802    1805    1808    1809    1810    1821    1822    1829    1831    1834    1837    1840    1843    1849    1851
                1854    1857    1860    1863    1864    1865    1876    1877    1884    1886    1889    1892    1895    1898    1904
                1906    1909    1912    1915    1918    1919    1920    1931    1932    1938    1941    1944    1947    1950    1953
                1959    1961    1964    1967    1970    1973    1974    1975    1986    1987    1994    1996    1999    2002    2005
                2008    2014    2016    2019    2022    2025    2028    2029    2030    2041    2042    2049    2051    2054    2057
                2060    2063    2069    2071    2074    2077    2080    2083    2084    2085    2096    2097    2104    2106    2109
                2112    2115    2118    2124    2126    2129    2132    2135    2138    2139    2140    2151    2152    2159    2161
                2164    2167    2170    2173    2179    2181    2184    2187    2190    2193    2194    2195    2206    2207    2214
                2216    2219    2222    2225    2228    2234    2236    2239    2242    2245    2248    2249    2250    2261    2262
                2269    2271    2274    2277    2280    2283    2289    2291    2294    2297    2300    2303    2304    2305    2316
                2317    2324    2326    2329    2332    2335    2338    2344    2346    2349    2352    2355    2358    2359    2360
                2371    2372    2379    2381    2384    2387    2390    2393    2399    2401    2404    2407    2410    2413    2414
                2415    2426    2427    2434    2436    2439    2442    2445    2448    2454    2456    2459    2462    2465    2468
                2469    2470    2481    2482    2489    2491    2494    2497    2500    2503    2509    2511    2514    2517    2520
                2523    2524    2526    2527    2538    2539    2546    2548    2551    2554    2557    2560    2566    2568    2571
                2574    2577    2580    2582    2583    2594    2595    2602    2604    2607    2610    2613    2616    2622    2624
                2627    2630    2633    2636    2638    2639    2650    2651    2658    2660    2663    2666    2669    2672    2678
```

```
        6235  6237  6253  6264  6266  6268  6271  6273  6276  6284  6287  6289  6303  6307  6310
        6313  6316  6322  6325  6331  6335  6338  6342  6345  6346  6348  6364  6375  6377  6379
        6382  6384  6387  6395  6398  6400  6414  6418  6421  6424  6427  6433  6436  6442  6446
        6449  6453  6456  6460  6476  6488  6490  6492  6495  6497  6500  6503  6509  6512  6514
        6529  6533  6536  6540  6543  6550  6553  6558  6562  6565  6569  6572  6574  6590  6602
        6604  6606  6609  6611  6614  6617  6623  6626  6628  6643  6647  6650  6654  6657  6664
        6667  6672  6676  6679  6683  6686  6688  6704  6716  6718  6720  6723  6725  6728  6731
        6737  6740  6742  6757  6761  6764  6768  6771  6778  6781  6786  6790  6793  6797  6800
        6802  6818  6830  6832  6834  6837  6839  6842  6845  6851  6854  6856  6871  6875  6878
        6882  6885  6892  6895  6900  6904  6907  6911  6914  6931  6944  6999  7001  7013  7015
        7018  7021  7024  7027  7031  7034  7046  7053  7096  7098  7111  7115  7117  7120  7123
        7127  7130  7133  7136  7148  7155  7199  7201  7206  7209  7221  7224  7227  7230  7233
        7245  7252  7292  7294  7299  7304  7308  7311  7322  7325  7328  7332  7335  7347  7354
        7394  7396  7401  7404  7411  7414  7426  7429  7432  7436  7439  7451  7458  7466  7473
        7481  7497  7500  7506  7509  7641  7766  7960  8199  8207  8208  8209  8210  8211  8212
        8213  8214  8215  8216  8217  8218  8219  8220  8446  8500  8504  8508  8736  8859  8922
        8946
.PAGE    266   402   629   736   822   984  1039  1094  1149  1204  1259  1314  1369  1424  1479
        1534  1589  1644  1699  1754  1809  1864  1919  1974  2029  2084  2139  2194  2249  2304
        2359  2414  2469  2526  2582  2638  2694  2750  2806  2862  2918  2974  3029  3084  3139
        3194  3249  3304  3359  3414  4143  4196  4249  4302  4355  4408  5195  5478  5523  5580
        5637  5694  5751  5808  5947  6013  6124  6235  6346
.PSECT    14
.RADIX   984   997  1039  1052  1094  1107  1149  1162  1204  1217  1259  1272  1314  1327  1369
        1382  1424  1437  1479  1492  1534  1547  1589  1602  1644  1657  1699  1712  1754  1767
        1809  1822  1864  1877  1919  1932  1974  1987  2029  2042  2084  2097  2139  2152  2194
        2207  2249  2262  2304  2317  2359  2372  2414  2427  2469  2482  2524  2526  2539  2582
        2595  2638  2651  2694  2707  2750  2763  2806  2819  2862  2875  2918  2931  2974  2987
        3029  3042  3084  3097  3139  3152  3194  3207  3249  3262  3304  3317  3359  3372  3414
        3427  3469
.REM       1    14    30
.REPT     96   305   306  1644  3029
.SBTTL    79    90   109   232   243   266   314   402   628   631   733   734   735   738   771
         823   862   910   948   985  1040  1095  1150  1205  1260  1315  1370  1425  1480  1535
        1590  1645  1700  1755  1810  1865  1920  1975  2030  2085  2140  2195  2250  2305  2360
        2415  2470  2527  2583  2639  2695  2751  2807  2863  2919  2975  3030  3085  3140  3195
        3250  3305  3360  3415  3469  3470  3471  3474  3519  3569  3620  3666  3716  3767  3809
        3855  3914  3994  3995  3996  3999  4067  4145  4198  4251  4304  4357  4410  4463  4519
        4575  4626  4690  4744  4813  4884  4932  4981  5030  5138  5195  5196  5197  5200  5269
        5356  5433  5478  5525  5582  5639  5696  5753  581C  5867  5943  5944  5945  5949  6015
        6126  6237  6348  6460  6574  6688  6802  6917  6919  6955  6956  6957  6961  7057  7159
        7256  7358  7511  7512  7513  7515  7592  7642  7689  7756  7820  7858  7997  8076  8133
        8176  8199
.TITLE    68
.WORD     96    97    98   240   259   260   261   262   263   264   274   277   278   279   280
         283   284   285   286   287   288   289   292   293   294   303   305   306   319   320
         321   322   323   324   325   326   330   331   332   345   349   352   355   356   357
         358   359   360   361   362   363   364   365   366   367   368   369   370   371   372
         373   374   375   376   377   603   606   608   610   612   614   616   618   620   621
         623   625   759   761   763   765   767   769   805   807   844   846   889   891   908
         928   930   965   967  1004  1006  1024  1026  1059  1061  1079  1081  1114  1116  1134
        1136  1169  1171  1189  1191  1224  1226  1244  1246  1279  1281  1299  1301  1334  1336
        1354  1356  1389  1391  1409  1411  1444  1446  1464  1466  1499  1501  1519  1521  1554
        1556  1574  1576  1609  1611  1629  1631  1664  1666  1684  1686  1719  1721  1739  1741
```

```
1774  1776  1794  1796  1829  1831  1849  1851  1884  1886  1904  1906  1939  1941  1959
1961  1994  1996  2014  2016  2049  2051  2069  2071  2104  2106  2124  2126  2159  2161
2179  2181  2214  2216  2234  2236  2269  2271  2289  2291  2324  2326  2344  2346  2379
2381  2399  2401  2434  2436  2454  2456  2489  2491  2509  2511  2546  2548  2566  2568
2602  2604  2622  2624  2658  2660  2678  2680  2714  2716  2734  2736  2770  2772  2790
2792  2826  2828  2846  2848  2882  2884  2902  2904  2938  2940  2958  2960  2994  2996
3014  3016  3049  3051  3069  3071  3104  3106  3124  3126  3159  3161  3179  3181  3214
3216  3234  3236  3269  3271  3289  3291  3324  3326  3344  3346  3379  3381  3399  3401
3434  3436  3454  3456  3496  3498  3503  3542  3548  3550  3588  3598  3600  3642  3644
3649  3685  3693  3697  3735  3743  3747  3781  3783  3788  3792  3827  3829  3832  3834
3853  3876  3878  3885  3887  3890  3894  3944  3946  3954  3959  3962  3968  4017  4020
4025  4027  4030  4034  4051  4085  4094  4097  4099  4102  4106  4123  4164  4166  4169
4175  4181  4217  4219  4222  4228  4234  4270  4272  4275  4281  4287  4323  4325  4328
4334  4340  4376  4378  4381  4387  4393  4429  4431  4434  4440  4446  4483  4485  4488
4495  4500  4540  4544  4546  4549  4554  4559  4598  4601  4604  4606  4609  4611  4653
4656  4659  4661  4664  4666  4709  4714  4722  4724  4727  4730  4768  4777  4780  4783
4786  4788  4791  4796  4837  4846  4849  4852  4855  4857  4860  4865  4904  4907  4910
4912  4915  4917  4952  4955  4958  4960  4963  4965  5005  5008  5010  5013  5015  5072
5074  5077  5080  5083  5085  5088  5095  5116  5137  5160  5163  5166  5169  5175  5179
5195  5224  5229  5234  5237  5239  5242  5244  5295  5299  5306  5312  5315  5321  5324
5329  5383  5385  5388  5391  5394  5396  5399  5401  5417  5452  5454  5457  5464  5492
5494  5496  5499  5507  5545  5547  5549  5552  5560  5565  5602  5604  5606  5609  5617
5622  5659  5661  5663  5666  5674  5679  5716  5718  5720  5723  5731  5736  5773  5775
5777  5780  5788  5793  5830  5832  5834  5837  5845  5850  5898  5900  5902  5905  5908
5911  5914  5916  5919  5921  5980  5983  5985  5988  5993  6042  6044  6046  6049  6051
6054  6062  6065  6067  6081  6100  6103  6109  6153  6155  6157  6160  6162  6165  6173
6176  6178  6192  6211  6214  6220  6264  6266  6268  6271  6273  6276  6284  6287  6289
6303  6322  6325  6331  6375  6377  6379  6382  6384  6387  6395  6398  6400  6414  6433
6436  6442  6488  6490  6492  6495  6497  6500  6503  6509  6512  6514  6529  6550  6553
6558  6602  6604  6606  6609  6611  6614  6617  6623  6626  6628  6643  6664  6667  6672
6716  6718  6720  6723  6725  6728  6731  6737  6740  6742  6757  6778  6781  6786  6830
6832  6834  6837  6839  6842  6845  6851  6854  6856  6871  6892  6895  6900  6936  6939
6951  6999  7001  7013  7015  7018  7021  7096  7098  7111  7115  7117  7120  7123  7199
7201  7206  7209  7221  7292  7294  7299  7304  7308  7311  7322  7394  7396  7401  7404
7411  7414  7426  7509  7510  7591  7668  7673  7857  8026  8073  8108  8168  8206  8379
8381  8383  8385  8387  8389  8391  8393  8394  8395  8397  8399  8401  8403  8405  8452
8549  8569  8704  8770  8771  8778  8813  8823  8844  8845  8859  8869  8870  8900  8904
8934  8935  8940  8951  8952  8955  8956  8986  8987  8993  8994  9000
```

000000

ERRORS DETECTED: 0

```
*DRLPG,DRLPG/SOL/CRF=DRLPA.MAC,DRLPG
RUN-TIME:  47 42 5 SECONDS
CORE USED:  47K
```