KWV11-A

DIAGNOSTIC

MD-11-DVKWA-B

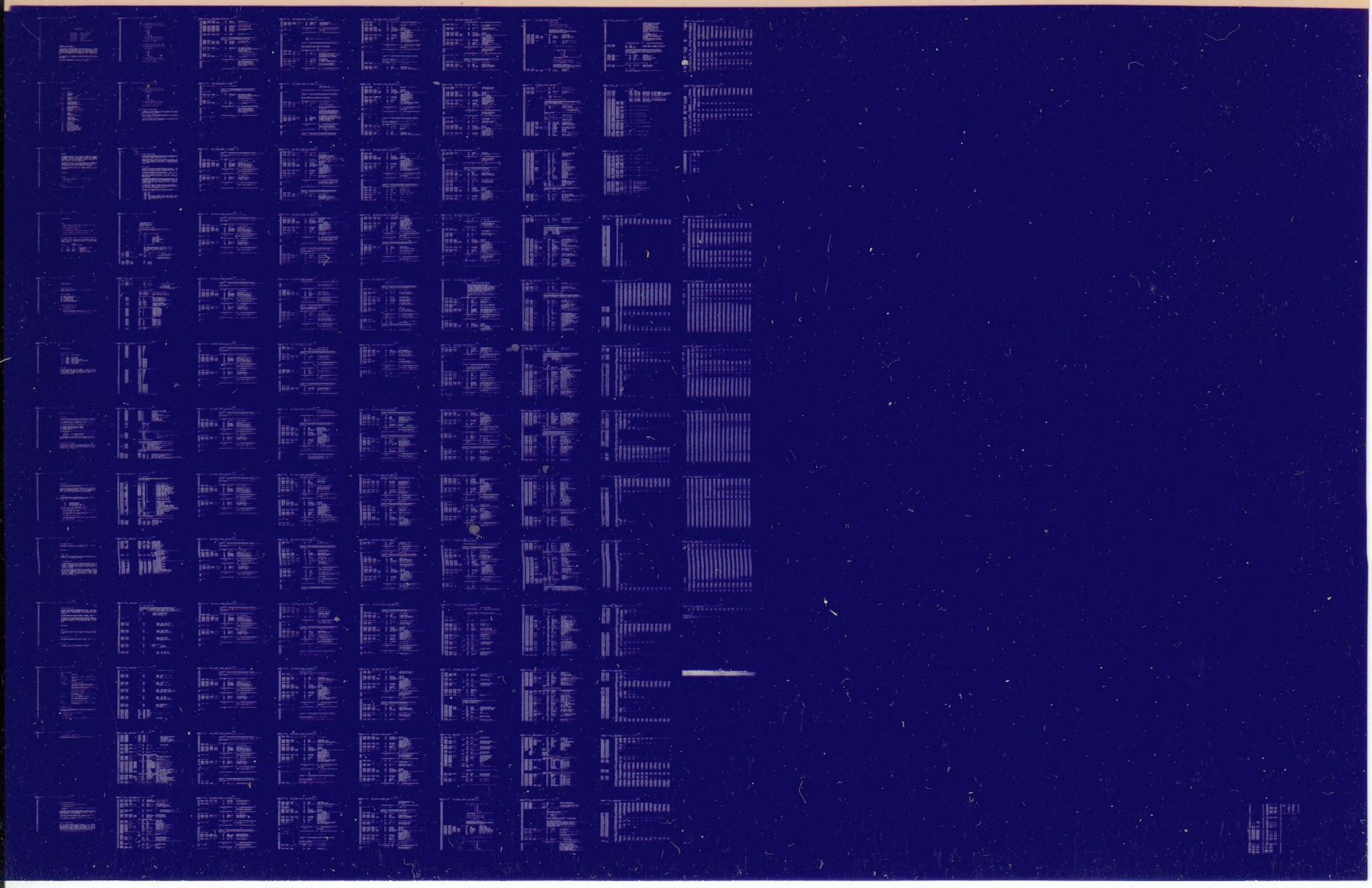
EP DVKWA B DL B

FICHE 1 OF 1

MAR 1977

digital

MADE IN USA



```
                                .REM %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DVKWA-B-D
PRODUCT NAME:	KWV11A DIAGNOSTIC
DATE CREATED:	OCTOBER 1976
DATE REVISED:	JANUARY 1977
MAINTAINER:	DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM ALLOWS THE USER CHECK-OUT OR DEBUG THE KWV11A, PROGRAMMABLE REAL-TIME CLOCK. THE LOGIC TEST IS SELF CONTAINED AND NEEDS NO EXTERNAL MAINTENANCE HARDWARE OR OPERATOR INTERVENTION WITH ONLY ONE EXCEPTION: IF THE CUSTOMER HARDWARE CONNECTED TO THE KWV11 COULD INJECT SIGNALS ON ST2, ST1, OR SLAVE IN INPUTS, IT MUST BE DISCONNECTED.

EVEN THOUGH THE KWV11 IS A Q BUS OPTION, THIS PROGRAM WAS DESIGNED TO RUN ON ANY PDP-11 FAMILY COMPUTER. IF THE USER IS UNFAMILAR WITH AN LSI-11 HE SHOULD REVIEW SECTIONS 8.4 AND 8.5. A SOFTWARE SWITCH REGISTER IS INCLUDED WITH THIS PROGRAM.

EVERY EFFORT WAS MADE TO MAKE THIS PROGRAM CONFORM TO LSI-11 PROGRAMMING RESTRICTIONS, HOWEVER; THE USER SHOULD READ SECTIONS 7.2 AND 7.3.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP-11 FAMILY COMPUTER WITH 8K OF MEMORY (OR MORE) AND I/O FACILITIES (I.E., TTY).

2. KWV11 UNDER TEST.

2.2 STORAGE

THIS PROGRAM OCCUPIES AND USES 8K OF MEMORY.

3.0 LOADING PROCEDURE

3.1 METHOD

STANDARDS PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

1. ABSOLUTE LOADER MUST BE IN MEMORY.
2. PLACE BINARY TAPE IN READER.
3. TYPE ADDRESS *7500 (5* DETERMINE BY LOCATION OF LOADER).
4. TYPE "G" (PROGRAM WILL BE LOADED INTO MEMORY).

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

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

THIS PROGRAM IS SET TO TEST A KWV11 WITH A STANDARD ADDRESS AND VECTOR. 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
1250	$BASE:	170420	;;BASE ADDRESS OF EQUIPMENT ;; UNDER TEST
1244	$VECT1:	000440	;;INTERRUPT VECTOR #1
176	$SWREG:	000000	;;MANUAL SWR.
1157	$TPFLG:	.BYTE 0	;;"TERMINAL AVAILABLE" ;; FLAG (BIT<0:7>=0=YES)

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

BEFORE STARTING THE DIAGNOSTIC, SET ALL SWITCH REGISTER BITS AS DESIRED, SEE SECTION 5.1.

4.2 STARTING ADDRESSES

200 START OF LOGIC TESTS
204 RESTART ADDRESS FOR LOGIC TEST
210 I/O SIGNAL TEST #1
214 I/O SIGNAL TEST #2
220 I/O SIGNAL TEST #3
230 PRODUCTION STARTING ADDRESS
240 TESTOR STARTING ADDRESS

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY.
2. ENTER KEYBOARD "ODT".
3. ALTER LOCATION "$SWREG" (ADDRESS 176) TO REFLECT DESIRED OPTIONS OF A SWITCH REGISTER - SEE SECTION 5.1.
4. TYPE STARTING ADDRESS, FOLLOWED BY "G" TO START PROGRAM.

5.0 OPERATING PROCEDURE

5.1 SWITCH REGISTER FUNCTION

SWR BIT	OCTAL	FUNCTION WHEN SET
15	100000	HALT ON ERROR
14	040000	LOOP ON TEST
13	020000	INHIBIT ERROR TYPEOUT
12	010000	ENABLE LINE FREQ. RATE TESTING
11	004000	INHIBIT ITERATIONS (SHORT PASS)
10	002000	BELL ON ERROR
09	001000	LOOP ON ERROR
08	000400	LOOP ON TEST IN SWR <7:0>

5.2 SCOPE LOOPS

IF AN ERROR OCCURS AND THE USER WISHES TO SCOPE THE ERROR, "$SWREG" SHOULD BE ALTERED TO "100000" AT THE START OF THE TEST TO HALT ON ERROR, THEN WHEN THE PROGRAM HALTS ON ERROR AND THE CPU ENTERS "ODT", "$SWREG" SHOULD BE ALTERED TO "060000" TO LOOP ON CURRENT TEST AND INHIBIT ERROR TYPEOUT, THEN TYPE "P" TO CONTINUE PROGRAM EXECUTION.

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.

IF NOT INHIBITED BY APT, THE PROGRAM WILL LOOK FOR MORE KWV11'S TO EXERCISE, ONE PASS WILL EXERCISE ALL KWV11'S.

IF FOUR UNITS ARE DETECTED, THE FOLLOWING WILL BE TYPED:

```
UNIT #000001 COMPLETED TESTING UNIT #000002
UNIT #000002 COMPLETED TESTING UNIT #000003
UNIT #000003 COMPLETED TESTING UNIT #000004
UNIT #000004 COMPLETED
```

AT END OF PASS WHEN ALL UNITS HAVE BEEN TESTED, THE FOLLOWING TYPEOUT WILL OCCUR:

"ENDPASS 12 - TOTAL ERRORS 4 THERE ARE 4 (OCTAL) UNITS - GOOD UNITS (L TO R) 0000000000001011".

THIS INDICATES THAT THE PROGRAM HAS COMPLETED 12 OCTAL (10 DECIMAL) PASSES. DURING THAT TIME 4(OCTAL) ERRORS WERE DETECTED. ALSO WE TESTED 4 UNITS AND THE THIRD UNIT WAS THE ONLY UNIT TO FAIL.

5.4 INHIBITING AUTO-SIZE FEATURE

THIS PROGRAM WILL AUTOMATICALLY AUTO-SIZE AND TEST EACH KWV11 IT DETECTS ON THE SYSTEM. TO INHIBIT THIS FEATURE, SET BIT 15 OF LOCATION "$ENVM". ALSO, TO TEST AN INDIVIDUAL KWV11 IN A GROUP, SET THIS BIT AND REFER TO SECTION 3.2 FOR CHANGING THE BASE ADDRESS OF THE KWV11 UNDER TEST.

6.0 ERRORS

6.1 ERROR PRINTOUT

PRINTOUT VARIES WITH THE ERROR DETECTED. THE ERROR PC TYPED OUT IS THE 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 TYPEOUT WOULD HAVE BEEN GOTO TO THE ERROR POINTER TABLE BEGINNING AT LOCATION "ERRTB".

6.1.1 EXAMPLE

IF WE EXAMINED LOCATION "$TSTNM" AND FOUND A 5(10) WE GO TO LOCATION "$ERRTB" AND LOOK THROUGH THE ERROR POINTER TABLE UNTIL WE FOUND ITEM 5. THE INFORMATION WOULD LOOK LIKE:

```
;ITEM 5

        EM5     ;CLOCK SR DATA ERROR
        DH5     ;ERRPC ASR WAS S/B
        DT5     ;$ERRPC,ASR,$BDDAT,$GDDAT
        DF0     ;ALL NUMBERS ARE IN OCTAL FORM
```

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

1. LOOK UP THE ADDRESS OF THE LABEL (I.E., $ERRPC) IN THE SYMBOL TABLE WHICH FOLLOWS THE LISTING.

2. * PUT THIS ADDRESS IN THE SWITCH 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.

* SEE SECTION 8.4 FOR LSI-11 ODT COMMANDS.

6.2 NON-STANDARD ERROR HALTS

BUS ERRORS WILL CAUSE A HALT TO THE ROUTINE "IOTRD". THE ADDRESS THAT CAUSED THIS TRAP WILL BE IN ADDRESS "TRTO".

7.0 RESTRICTIONS

7.1 EXTERNAL INPUTS

EXTERNAL INPUTS SUCH AS "SLAVE IN", "ST1" AND "ST2" MUST NOT BE CONNECTED TO ANY CUSTOMER HARDWARE THAT MIGHT GENERATE THESE SIGNAL WHILE THE DIAGNOSTIC IS RUNNING.

7.2 STARTING RESTRICTION

IF A FREE-RUNNING CLOCK, SUCH AS 60HZ FROM THE POWER SUPPLY, IS ATTACHED TO THE "BEVNT" BUS LINE ON BOTH REV LEVEL C/D AND E SYSTEMS, AN INTERRUPT TO LOCATION 100 WILL OCCUR WHEN USING THE "G" AND "L" COMMANDS PRIOR TO EXECUTING THE FIRST INSTRUCTION. THEREFORE THIS PROGRAM CAN NOT DISABLE THE BEVNT BUS LINE BY INHIBITING INTERRUPTS.

USER SYSTEMS REQUIRING A FREE-RUNNING CLOCK ATTACHED TO THE BEVNT BUS LINE CAN TEMPORARILY AVOID THIS SITUATION BY SETTING THE PSW(RS) TO 200, LOADING THE PC WITH THE STARTING ADDRESS INSTEAD OF USING THE "G" COMMAND, AND THEN USING THE "P" COMMAND. BEFORE USING THE "L" COMMAND, THE PSW(RS) CAN BE SET TO 200, THEREBY INHIBITING INTERRUPTS, TO AVOID RECEIVING THE EVENT INTERRUPT AFTER LOADING THE ABS LOADER.

7.3 POSSIBLE PROGRAM "BOMBS"

THE FIRST TWO TESTS OF THIS PROGRAM CHECK TO SEE IF THE KWV11 RESPONDS TO THE ADDRESS THE PROGRAM THINKS ITS AT. IF THE KWV11 DOES NOT RESPOND, A BUS ERROR OCCURS. ALSO BUS ERRORS CAN OCCUR DURING THE TIME THE PROGRAM SIZES TO SEE HOW MANY KWV11 ARE ON YOU SYSTEM.

FOR MORE INFORMATION ON THE NEXT SUBJECT, SEE JAN. 1976 LSI-11 ENGINEERING BULLETIN ISSUED BY THE DIGITAL COMPONENTS GROUP.

BUS ERRORS MAY ALTER THE PRESET CONTENTS OF LOCATION 4 BEFORE THE TRAP IS EXECUTED, THEREBY TRANSFERRING PROGRAM CONTROL TO AREA IN THE PROGRAM THAT WAS NOT SET UP TO HANDLE THE TRAP. IF THIS HAPPENS, THE PROGRAM WILL "BOMB" AND POSSIBLY REWITE PARTS OF ITSELF.

8.0 MISCELLANEOUS

8.1 POWER FAIL

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

8.2 XXDP, ACT, APT

THE PROGRAM IS CHAINABLE UNDER XXDP, ACT, OR APT. ALTHOUGH "APT HOOKS" HAVE BEEN INSTALLED, THEY HAVE NOT BEEN TESTED.

8.3 EXECUTION TIME

0.5 MINUTES (30 SEC) ITERATION INHIBITED - NO ERRORS
2.5 MINUTES (150 SEC) WITH ITERATIONS - NO ERRORS

8.4 LSI-11 "ODT" COMMANDS

FORMAT	DESCRIPTION
<CR> RETURN	CLOSE OPENED LOCATION AND ACCEPT NEXT COMMAND.
<LF> LINE FEED	CLOSE CURRENT LOCATION; OPEN NEXT SEQUENTIAL LOCATION.
↑(UPARROW)	OPEN PREVIOUS LOCATION.
← (LEFT ARROW)	TAKE CONTENTS OF OPENED LOCATION, INDEXED BY CONTENTS OF PC, AND OPEN THAT LOCATION.
@	TAKE CONTENTS OF OPENED LOCATION AS ABSOLUTE ADDRESS AND OPEN THAT LOCATION.
R/	OPEN THE WORD AT LOCATION R.
/	REOPEN THE LAST LOCATION.
$N/ OR RN/	OPEN GENERAL REGISTER N(0-7) OR S(PS REGISTER).
R;G OR RG	GOTO LOCATION R AND START PROGRAM.
NL	EXECUTE BOOTSTRAP LOADER USING N AS DEVICE CSR. CONSOLE DEVICE IS 177560.
;P OR P	PROCEED WITH PROGRAM EXECUTION.
RUBOUT	ERASES PREVIOUS NUMERIC CHARACTER. RESPONSE IS A BACKSLASH ().

8.5 ENTERING LSI-11 "ODT"

THE HALT OR ODT MICROCODE STATE OF THE KD11F (LSI-11 MODULE) CAN BE ENTERED IN FIVE DIFFERENT WAYS (OTHERS ARE A SUBSET OF THESE) FROM THE RUN STATE:

1. EXECUTION OF A LSI-11 HALT INSTRUCTION,
2. A DOUBLE BUS ERROR,
3. AS A POWER UP OPTION,
4. ASCII BREAK WITH DLV11 FRAMING ERROR ASSERTING THE B HALT

```
                                    LINE (ENABLED BY JUMPER OF DLV11).

                                UPON ENTERING THE HALT STATE, THE KD11F RESPONDS THROUGH THE  SET
                                OF COMMAND LISTED IN SECTION 8.4.
```

8.6 USE OF PROGRAM SOFTWARE SWR

THE PROGRAM SOFTWARE SWITCH REGISTER IS ENABLED IF

1. NO HARDWARE SWR EXISTS;

2. IF YOU START WITH ALL ONES (SWR=177777) IN THE SWITCH REGISTER.

THE SOFTWARE SWITCH REGISTER MAY BE CHANGED BY TYPING ↑G (CONTROL AND LETTER G KEYS TYPED SIMULTANEOUSLY). WHEN ↑G IS TYPED, THE PROGRAM RESPONDS BY TYPING "SWR=XXXXXX" WHERE XXXXXX EQUALS THE FORMER CONTENTS OF THE SWITCH REGISTER.

IF YOU WISH TO KEEP THE CURRENT VALUE, TYPE <CR>. IF YOU WISH TO CHANGE THE VALUE, TYPE THE NEW VALUE FOLLOWED BY A <CR>.

IT IS IMPORTANT TO NOTE THAT THE DIAGNOSTIC IS NOT RUNNING AFTER THE ↑G UNTIL A <CR> IS TYPED.

8.7 SPECIAL I/O SIGNAL TESTS

THREE TESTS WERE INCLUDED TO ENABLE CHECKOUT OF I/O SIGNALS: ST1, ST2, AND CLOCK OVERFLOW. THESE TESTS HAVE A SPECIAL STARTING ADDRESS. SINCE END-PASSES ARE IMMEDIATE, NO "END OF PASS" MESSAGE IS REPORTED. ERRORS ARE REPORTED BY TYPING OUT THE PC WHERE THE ERROR WAS DETECTED. WHEN STARTED, THE PROGRAM REMAINS IN A LOOP GENERATING AND DETECTING THE SPECIFIED SIGNALS. HALT ON ERROR AND INHIBIT ERROR TYPEOUT OPTIONS MAY BE USED.

LOGIC TEST MUST HAVE ALREADY BEEN RUN ON THE KWV11.

8.7.1 I/O SIGNAL TEST #1 ST1 IN, ST2 OUT

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH	STATE
1	OFF
2	ON
3	OFF
4	OFF
5	ON
6	ON
7	NOT USED

THE FOLLOWING JUMPER MUST BE INSTALLED.

J1-SS (ST2 OUT) TO J1-VV (ST1 IN)

LOAD AND START THE PROGRAM AT 210.

8.7.2 I/O SIGNAL TEST #2 CLOCK OVERFLOW TEST

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH	STATE
1	OFF
2	OFF
3	OFF
4	ON
5	ON
6	OFF
7	NOT USED

THE FOLLOWING JUMPER MUST BE INSTALLED.

J1-RR (CLOCK OVERFLOW) TO J1-TT (ST2 IN)

LOAD AND START AT LOCATION 214.

8.7.3 I/O SIGNAL TEST #3 ST1 OUT AND ST2 IN

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH	STATE
1	OFF
2	OFF
3	OFF
4	ON
5	ON
6	ON
7	NOT USED

THE FOLLOWING JUMPER MUST BE INSTALLED:

J1-UU (ST1 OUT) TO J1-TT (ST2 IN)

LOAD AND START AT LOCATION 220.

8.8 PRODUCTION STARTING ADDRESS

A SPECIAL STARTING ADDRESS HAS BEEN PROVIDED FOR IN-HOUSE PRODUCTION TO USE TO START THE LOGIC DIAGNOSTIC AND INFORM THE TEST THAT PRODUCTION IS USING IT.

IN THE FIELD ONLY ENOUGH ADDRESSES WERE ALLOTED FOR 4 SEQUENTIAL KWV11S. WHEN THE LOGIC TESTS ARE STARTED AT LOCATION 200, WE ONLY AUTO-SIZE UP TO 4 KWV11S.

IN HOUSE TESTING MAY WISH TO EXERCISE UP TO 16 KWV11S AT ONE TIME. THE LOGIC TESTS MAY BE STARTED AT LOCATION 230 AND THE PROGRAM WILL AUTO SIZE UP TO 16 KWV11S.

8.9 TESTOR STARTING ADDRESS

A SPECIAL STARTING ADDRESS HAS BEEN PROVIDED FOR MANUFACTURING TO USE TO START THE LOGIC DIAGNOSTIC AND INFORM THE PROGRAM THAT THE CLOCK MODULE IS CABLED TO AN IN-HOUSE TESTOR.

MANUAL INTERVENTION IS NEEDED IN THIS SEQUENCE OF TESTING. THE PROGRAM WILL TYPE OUT ALL INSTRUCTIONS. A CABLE SHOULD CONNECT J1 ON THE CLOCK MODULE TO J10 ON THE TESTOR. SWITCHES 1 AND 3 OF S2 (ON THE CLOCK MODULE) SHOULD BE ON, ALL OTHER SWITCHES ON S2 SHOULD BE OFF.

8.10 TRAP CATCHER

THE TRAP CATCHER IN THIS DIAGNOSTIC EMPLOYS A NEW CONCEPT. THIS CONCEPT WILL ENABLE THE USER OF THIS DIAGNOSTIC TO GAIN MORE KNOWLEDGE OF THE EVENTS THAT LEAD THE PROGRAM TO THIS AREA.

THE TRAP CATCH CONSISTS OF PC+2 AND JSR PC,R0. (I.E., LOCATION 300 WOULD CONTAIN 302 AND LOCATION 302 WOULD CONTAIN 4700.)

WHEN A DEVICE INTERRUPTS UNEXPECTEDLY TO THE TRAP CATCHER, IT WOULD PICK UP THE PC+2 OF THE TRAP AS AN ADDRESS OF THE INTERRUPT SERVICE ROUTINE.

THE PROGRAM WOULD THEN PICK UP "4700" AS THE NEW PSW. BIT 7 OF THE NEW PSW HAVING BEEN SET, WOULD CAUSE FURTHER INTERRUPTS FROM HAPPENING. WHEN THE CPU ATTEMPTS TO EXECUTE "4700" (JSR PC,R0), A BUSS-TIME-OUT TRAP WILL OCCUR TO LOCATION 4. LOCATION 4 CONTAINS A POINTER TO "IOTRD", A ROUTINE THAT WILL REPORT THE TRAP AS AN ERROR.

TO GUARD AGAINST "REAL" BUS ERRORS ROUTING US THROUGH LOCATION 4 TO "IOTRD", WE CHECK TO SEE IF THE TRAP THAT BROUGHT US TO LOCATION 4 REALLY CAME FROM THE TRAP CATCHER AREA. IF NOT WE'LL HALT AND LEAVE THE TRAP ADDRESS IN "TRTO".

MORE ABOUT THE INTERRUPT ERROR CAN BE FOUND IN THE DESCRIPTION OF THE ERROR IN THE PROGRAM LISTING IN THE ROUTINE "IOTRD".

```
                        %
                        .NLIST  MC,MD,CND
                        .LIST   ME
                        .ENABL  ABS
                        .ENABL  AMA
                        .MCALL  .HEADER,.SETUP,.SETTRAP,.TRMTRP,.$TRAP,.$RDOCT,.$TYPBIN
                        .MCALL  TYPOCS,.$POWER,.$CATCH,.$TYPOCT,.EQUAT,.$CMTAG,.SWRHI
                        .MCALL  .$EOP,.$ERROR,.$ERRTYP,.$TYPDEC,.$SCOPE,.$READ,.$TYPE
                        .MCALL  .$ACT11,.$     ?R,.$APTYPE
167400                  $SWR=   167400
```

```
                                .TITLE  MAINDEC-11-DVKWA-B
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY EDWARD C. BADGER
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-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

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2"
                                ;*AND "JSR PC,R0" SEQUENCE TO CATCH ILLEGAL INTERRUPTS.
                                ;*AND INTERRUPTS TO THE WRONG VECTOR.
                                ;*LOCATION 0 CONTAINS A 0 TO CATCH IMPROPERLY LOADED
                                ;*VECTORS.
000004                          .=4
000004  017004  000200          .WORD   IOTRD,200       ;HANDLE BUSS ERROR.
000174                          .=174
000174  000000          DISPREG:        .WORD   0               ;;SOFTWARE DISPLAY REGISTER.
000176  000000          SWREG:  .WORD   0               ;;SOFTWARE SWITCH REGISTER.
000100                          .=100
000100  000104  000200  000002          .WORD   104,200,2       ;IF "B EVENT"ON Q-BUS IS
                                                                ;CONNECTED,WE NEED A WAY OF
                                                                ;IGNORING ITS INTERRUPTS.

000200                          .=200
000200  000137  001474          JMP     @#START
000204  000137  002104          JMP     @#RSTART
000210  000137  014006          JMP     @#IOTST1
000214  000137  014064          JMP     @#IOTST2
```

```
                                                JMP     @#IOTST3

                        000230                  .=230
                                                JMP     @#WSTART                ;WESTFIELD STARTING ADDRESS
                        000240                  .=240
                                                JMP     @#TSTSTR                ;ALL TESTER TESTS
                                                                                ;IF STARTED HERE.
                                                                                ;ALLOWS PRODUCTION TO EXERCISE
                                                                                ;UP TO 16 CLOCKS.NORMAL=4.

                                        .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

                        170420                          ABASE=  170420
                        000440                          AVECT1= 440
                        000200                          APRIOR= 200

                        167400                          $SWR=   167400
                        000001                          $TN=    1

                                                .SBTTL  ACT11 HOOKS

                                                ;;**************************************************************
                                                ;HOOKS REQUIRED BY ACT11
                        000244                          $SVPC=.                 ;SAVE PC
                        000046                          .=46
000046  013660                                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052                          .=52
000052  000000                                  .WORD   0               ;;2)SET LOC.52 TO ZERO
                        000244                          .=$SVPC                 ;; RESTORE PC
                        001000                          .=1000
                                                .SBTTL  APT PARAMETER BLOCK

                                                ;;**************************************************************
                                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                                ;;**************************************************************
                        001000                          .$X=.   ;;SAVE CURRENT LOCATION
                        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                                  200     ;;FOR APT START UP
                        000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                                  $APTHDR ;;POINT TO APT HEADER BLOCK
                        001000                          .=.$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  001174                                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000002                                  $TSTM:  .WORD   2       ;;RUN TIM OF LONGEST TEST
001006  000170                                  $PASTM: .WORD   120.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000170                                  $UNITM: .WORD   120.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000031                                          .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                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001162  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001164  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001170  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
001171  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001172  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**********************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;**********************************************************************
                                .EVEN
001174                          $MAIL:                  ;;APT MAILBOX
001174  000000                  $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
001176  000000                  $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
001200  000000                  $TESTN: .WORD   ATESTN   ;;TEST NUMBER
001202  000000                  $PASS:  .WORD   APASS    ;;PASS COUNT
001204  000000                  $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
```

```
927  001206  000000                  $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
928  001210  000000                  $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
929  001212  000000                  $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
930  001214                          $ETABLE:                ;;APT ENVIRONMENT TABLE
931  001214     000                  $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
932  001215     000                  $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
933  001216  000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
934  001220  000000                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
935  001222  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
936                                  ;*                      BITS 15-11=CPU TYPE
937                                  ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
938                                  ;*                              11/70=06,PDQ=07,Q=10
939                                  ;*                      BIT 10=REAL TIME CLOCK
940                                  ;*                      BIT  9=FLOATING POINT PROCESSOR
941                                  ;*                      BIT  8=MEMORY MANAGEMENT
942  001224     000                  $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
943  001225     000                  $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
944                                  ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
945                                  ;*                              900 NSEC CORE=001
946                                  ;*                              300 NSEC BIPOLAR=002
947                                  ;*                              500 NSEC MOS=003
948  001226  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
949                                  ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
950  001230     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
951  001231     000                  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
952  001232  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
953  001234     000                  $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
954  001235     000                  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
955  001236  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
956  001240     000                  $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
957  001241     000                  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
958  001242  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
959  001244  000440                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
960  001246  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
961  001250  170420                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
962  001252  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
963  001254  000000                  $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
964  001256                          $ETEND:
965                                  .MEXIT
```

```
                                .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

001256                          $ERRTB:

                                ;ITEM   1

001256  017134                          EM1                     ;CLOCK   SR FUNCTION ERROR
001260  017457                          DH1                     ;ERRPC   ASR     WAS     S/B
001262  017666                          DT1                     ;$ERRPC,ASR,$BDDAT,$GDDAT
001264  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   2

001266  017166                          EM2                     ;CLOCK   SR DATA ERROR
001270  017457                          DH1                     ;ERRPC   ASR     WAS     S/B
001272  017666                          DT1                     ;$ERRPC,ASR,$BDDAT,$GDDAT
001274  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   3

001276  017214                          EM3                     ;CLOCK  BR DATA ERROR
001300  017503                          DH3                     ;ERRPC   ABR     WAS
001302  017700                          DT3                     ;$ERRPC,ABR,$BDDAT,$GDDAT
001304  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   4

001306  017242                          EM4             ;INTERRUPT ERROR.
001310  017527                          DH4A            ;ERRPC   TO      ROM ADDR.
001312  017712                          DT4             ;$ERRPC,        TRTO,TRFRO
001314  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM

                                ;ITEM   5

001316  017263                          EM5                     ;CLOCK   COUNT REG ERROR
001320  017457                          DH1                     ;ERRPC   ASR   WAS   S/B
001322  017666                          DT1                     ;$ERRPC, ACR, $BDDAT, $GDDAT
```

```
1020  001324  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
1021
1022
1023                                  ;ITEM   6
1024
1025  001326  017325                          EM12                    ;CLOCK   COUNT FUNCTION ERROR
1026  001330  017563                          DH12                    ;ERRPC   ASR
1027  001332  017722                          DT12                    ;ERRPC, ASR
1028  001334  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
1029
1030
1031                                  ;ITEM   7
1032
1033  001336  017354                          EM16                    ;CLOCK   INTERRUPT ERROR
1034  001340  017563                          DH12                    ;ERRPC   ASR
1035  001342  017722                          DT12                    ;$ERRPC, ASR
1036  001344  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
1037
1038
1039                                  ;ITEM   10
1040
1041  001346  017405                          EM20                    ;CLOCK   REPEATABILITY ERROR
1042  001350  017600                          DH20                    ;ERFOR   ASR   2ND CNT 1ST CNT  3RD CNT
1043  001352  017730                          DT20                    ;$ERRPC, ASR, $BDDAT, $GDDAT, $TMP0
1044  001354  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
1045
1046
1047                                  ;ITEM   11
1048
1049  001356  017306                          EM11                    ;CLOCK   COUNT ERROR
1050  001360  017457                          DH1                     ;ERRPC   ASR   WAS   S/B
1051  001362  017744                          DT22                    ;$ERRPC, ASR, $BDDAT, $TMP0
1052  001364  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
1053
1054
1055                                  ;ITEM   12
1056
1057  001366  017434                          EM26                    ;CLOCK ADDRESSING ERROR
1059  001370  017641                          DH26                    ;ERRPC   CLOCK ADDR.
1059  001372  017756                          DT26                    ;$ERRPC,$TMP0
1060  001374  017764                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM
1061
1062
1063  001376  170420          ASR:    .WORD   ABASE
1064  001400  170422          ABR:    .WORD   ABASE+2
1065  001402  000440          VECT1:  .WORD   AVECT1
1066  001404  000442          VECTP:  .WORD   AVECT1+2
1067  001406  000444          VECT2:  .WORD   AVECT1+4        ;VECTOR ADDR. OF ST2 INTRS.
1068  001410  000446          VECT2P: .WORD   AVECT1+6
1069  001412  000200          PRIOR:  .WORD   APRIOR
1070  001414  167774          DR:     .WORD   167774
1071  001416  167772          DR2:    .WORD   167772
1072  001420  000000          $TMP0:  .WORD   0               ;TEMP STORAGE.
1073  001422  000000          $TMP1:  .WORD   0               ;TMP STORAGE.
```

```
1074  001424  000000                  $TMP3:  .WORD   0
1075  001426  000000                  ROTATE: .WORD   0                       ;POINT TO DEVICE UNDER TEST.
1076  001430  000000                  UTEST:  .WORD   0                       ;KEEPS TRACK OF GOOD UNITS.
1077  001432  000000                  ERCNT:  .WORD   0                       ;COUNTS ERRORS.
1078  001434  000000                  MDEVCT: .WORD   0                       ;COUNTS DEVICES TESTED.
1079  001436  000000                  TSTCNT: .WORD   0                       ;MAX DEVICES TO BE TESTED.
1080  001440  000000                  EXS:    .WORD   0                       ;=0, NORMAL: =1 SPECIAL TESTOR START, BY L+S @ 2
1081  001442  000000                  LCNT:   .WORD   0                       ;TOTAL UNITS TESTED.
1082
1083
1084  001444  005237  001440          TSTSTR: INC     EXS                     ;SET FOR TESTOR.
1085  001450  012737  000020  001436          MOV     #16.,TSTCNT             ;ALLOW 16 UNITS
1086  001456  000413                  WSTART=.BR      1$
1087          001460
1088  001460  012737  000020  001436          MOV     #16.,TSTCNT     ;TEST UP TO 16 UNITS.
1089  001466  005037  001440                  CLR     EXS
1090  001472  000405                  START=. BR      1$
1091          001474
1092  001474  012737  000004  001436          MOV     #4,TSTCNT       ;TEST UP TO FOUR UNITS.
1093  001502  005037  001440                  CLR     EXS
1094  001506                          1$:
1095                                  .SBTTL  INITIALIZE THE COMMON TAGS
1096                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1097  001506  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1098  001512  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1099  001514  022706  001140                  CMP     #$WR,R6 ;;DONE?
1100  001520  001374                          BNE     .-6             ;;LOOP BACK IF NO
1101  001522  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1102                                  ;;INITIALIZE A FEW VECTORS
1103  001526  012737  015040  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1104  001534  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1105  001542  012737  014476  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1106  001550  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1107  001556  012737  017054  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1108  001564  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1109  001572  012737  016626  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1110  001600  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1111  001606  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1112  001612  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1113  001616  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1114  001624  012737  001624  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1115  001632  012737  001632  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1116                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1117                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1118  001640  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1119  001644  012737  001700  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1120  001652  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1121  001660  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1122  001666  022777  177777  177744          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1123  001674  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1124                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1125  001676  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1126  001700  012716  001706          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1127  001704  000002                          RTI
```

```
1128  001706  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1129  001714  012737  000174  001142          MOV     #DISPREG,DISPLAY
1130  001722  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1131
1132  001726  005037  001202                  CLR     $PASS           ;;CLEAR PASS COUNT
1133  001732  132737  000200  001215          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1134  001740  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1135  001742  012737  001216  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1136  001750                          67$:
1137
1138
1139  001750  012746  000340                  MOV     #340,-(SP)              ;SET CPU PRIORITY ON RETERN.
1140  001754  012746  001762                  MOV     #68$,-(SP)              ;SHOW RETURN ADDRESS.
1141  001760  000002                          RTI                             ;CAUSE A RETURN(PUTS STATUS IN STATUS REG.).
1142  001762                          68$:
1143
1144  001762  005037  001204                  CLR     $DEVCT          ;ZERO DEVICE COUNT.
1145  001766  012737  017004  000004          MOV     #IOTRD,@#ERRVEC ;FIX TRAP CATCHER.
1146  001774  013737  001244  001402          MOV     $VECT1,VECT1    ;NOW FIX VECTOR ADDR.
1147  002002  013737  001250  001376          MOV     $BASE,ASR       ;FIX ADDRESS OF CSR.
1148
1149                          .SBTTL  TYPE PROGRAM NAME
1150                          ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1151  002010  005227  177777                  INC     #-1             ;;FIRST TIME?
1152  002014  001033                          BNE     69$             ;;BRANCH IF NO
1153  002016  104401  002064                  TYPE    ,70$            ;;TYPE ASCIZ STRING
1154
1155  002022  005737  000042          .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1156  002026  001012                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1157  002030  123727  001214  000001          BNE     71$             ;;BRANCH IF YES
1158  002036  001406                          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1159  002040  023727  001140  000176          BEQ     71$             ;;BRANCH IF YES
1160  002046  001005                          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1161  002050  104406                          BNE     72$             ;;BRANCH IF NO
1162  002052  000403                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1163  002054  112737  000001  001134  71$:    BR      72$
1164  002062                          72$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1165  002062  000410                  ;;70$:  BR      69$             ;;GET OVER THE ASCIZ
1166                                  69$:    .ASCIZ  <CRLF>#MD11-DVKWA-B#<CRLF>
1167  002104                          RSTART:
1168  002104
1169  002104  005737  001440                  TST     $XS             ;TESTOR MODE ENABLED??
1170  002110  001441                          BEQ     1$              ;NO DON'T TYPE NEXT MESSAGE.
1171  002112  104401  002120                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1172  002116  000436                          BR      64$             ;;GET OVER THE ASCIZ
1173                                  ;;65$:  .ASCIZ  <15><12>#TESTOR MODE ENABLED--SEE DOCUMENTATION FOR INSTRUCTIONS.#
1174  002214                          64$:
1175  002214                          1$:
1176  002214  104401  002222                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1177  002220  000411                          BR      66$             ;;GET OVER THE ASCIZ
1178                                  ;;67$:  .ASCIZ  <15><12>#TEST RUNNING...#
1179  002244                          66$:
1180  002244  005037  001434                  CLR     MDEVCT          ;TESTING FIRST UNIT.
1181  002250  005037  001432                  CLR     ERCNT           ;NO ERRORS.
```

```
1182  002254  005037  001202                  CLR     $PASS           ;NO PASSES.
1183  002260  012737  000001  001426          MOV     #1,ROTATE       ;POINT TO FIRST UNIT.
1184  002266  013737  001426  001430          MOV     ROTATE,UTEST
1185  002274                          LOOP:                           ;COME HERE FOR NEXT UNIT,OR END PASS.
1186
1187  002274  042737  170000  001402          BIC     #170000,VECT1   ;CLEAR OUT PRIORITY BITS.
1188  002302  013737  001402  001404          MOV     VECT1,VECTP     ;NOW FIX VECTOR +2 ADDR.
1189  002310  062737  000002  001404          ADD     #2,VECTP
1190  002316  013737  001402  001406          MOV     VECT1,VECT2     ;LETS FIX ST2 VECTOR ADDR.
1191  002324  062737  000004  001406          ADD     #4,VECT2        ;ITS 4 GREATER THEN THE 1ST.
1192  002332  013737  001406  001410          MOV     VECT2,VECT2P    ;VECTOR +2 ADDR.
1193  002340  062737  000002  001410          ADD     #2,VECT2P
1194
1195
1196  002346  013737  001376  001400          MOV     ASR,ABR         ;FIX ADDR OF PRESET REG=
1197  002354  062737  000002  001400          ADD     #2,ABR          ;CSR + 2
1198
1199
1200                                  ;;**************************************************************
1201                                  ;*TEST 1         *TEST THE ADDRESSABILITY OF CLOCK CSR
1202                                  ;;**************************************************************
1203  002362  000240                  TST1:   NOP
1204  002364  012737  000050  001160          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
1205  002372  012737  002414  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
1206  002400  112737  000001  001102          MOVB    #1,$TSTNM       ;SET TEST #1.
1207  002406  012737  002414  001110          MOV     #1$,$LPERR
1208
1209
1210
1211  002414  013746  000004          1$:     MOV     @#ERRVEC,-(SP)  ;SAVE CONTENTS OF ADDRS 6.
1212  002420  012737  002434  000004          MOV     #2$,@#ERRVEC    ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
1213                                                                  ;WE TIME-OUT WHEN ADDRESSING THE KW11.
1214  002426  005777  176744                  TST     @ASR            ;ADDRESS THE CLOCK!
1215                                                                  ;IF CLOCK DOES NOT RETURN
1216                                                                  ;"BUS SSYN" THEN WE'LL TIME-OUT
1217
1218  002432  000406                  BR      3$              ;THE CLOCK WAS THERE! EXIT SUB-TEST.
1219  002434                          2$:
1220  002434  062706  000004                  ADD     #4,SP                   ;/ADD #4 TO STACK POINTER.
1221  002440  013737  001376  001420          MOV     ASR,$TMPO       ;FOR ERROR TYPEOUT.
1222
1223

                                  ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

1226  002446  104012                  ERROR   12              ;REPORT ERROR=CLOCK CSR FAILED TO RETURN
1227                                                          ;"BUS SSYN" WHEN ADDRESSED.
1228                                                          ;NOTE: IF PROGRAM HAS INCORRECT
1229                                                          ;ADDRESS THEN WE MIG NOT BE
1230                                                          ;TALKING TO THE CLOCK. MAKE SURE
1231                                                          ;OF CLOCK ADDRESS.
1232
1233

                                  ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

```
002450  012637  000004          3$:     MOV     (SP)+,@#ERRVEC


                                ;;*************************************************************
                                ;*TEST 2        *TEST THE ADDRESSABILITY OF CLOCK BUFFER REG.
                                ;;*************************************************************
002454  000004                  TST2:   SCOPE


002456  013746  000004          1$:     MOV     @#ERRVEC,-(SP)   ;SAVE CONTENTS OF ADDRS 6.
002462  012737  002476  000004          MOV     #2$,@#ERRVEC     ;SET TIME-OUT TRAP VECTOR TO HANDLER IN CASE.
                                                                 ;WE TIME-OUT WHEN ADDRESSING THE KW11.
002470  005777  176704                  TST     @ABR             ;ADDRESS THE CLOCK!
                                                                 ;IF CLOCK DOES NOT RETURN
                                                                 ;"BUS SSYN" THEN WE'LL TIME-OUT

002474  000406                          BR      3$               ;THE CLOCK WAS THERE! EXIT SUB-TEST.
002476                          2$:
002476  062706  000004                  ADD     #4,SP                     ;/ADD #4 TO STACK POINTER.
002502  013737  001400  001420          MOV     ABR,$TMP0        ;FOR ERROR TYPEOUT.


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

002510  104012                          ERROR   12               ;REPORT ERROR=CLOCK BUFFER REG. FAILED TO RETURN
                                                                 ;"BUS SSYN" WHEN ADDRESSED.
                                                                 ;NOTE: IF PROGRAM HAS INCORRECT
                                                                 ;ADDRESS THEN WE MIG NOT,BE
                                                                 ;TALKING TO THE CLOCK. MAKE SURE
                                                                 ;OF CLOCK ADDRESS.


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

002512  012637  000004          3$:     MOV     (SP)+,@#ERRVEC
```

```
                                                ;/#
                                        ;;*************************************************************
                                        ;*TEST 3        *TEST THAT CLOCK A STATUS REGISTER BIT 14 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*

                                        ;;*************************************************************
002516  000004                          TST3:   SCOPE
002520  012737  000100  001160                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS

002526  005077  176644                          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
002532  052777  040000  176636                  BIS     #BIT14,@ASR     ;/SET BIT 14.
002540  012737  040000  001124                  MOV     #BIT14,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
002546  017737  176624  001126                  MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTEF.
002554  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 14 AND ONLY BIT 14 SET?
002562  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

002564  104002                                  ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                                        ;/BIT 14 FAILED TO BIT SET.

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

002566  000412                                  BR      2$              ;/BR TO END SUBTEST.

002570  042777  040000  176600  1$:     BIC     #BIT14,@ASR     ;/TRY CLEARING BIT 14.
002576  005037  001124                          CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
002602  017737  176570  001126                  MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
002610  001401                                  BEQ     2$              ;/IF ZERO - NO ERROR!

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

002612  104002                                  ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                                        ;/BIT 14 FAILED TO CLEAR.

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

002614                                  2$:
```

```
                                                   ;/*
                                           ;;*************************************************************
                                           ;*TEST 4        *TEST THAT CLOCK A STATUS REGISTER BIT 13 CAN BE SET AND CLEARED
                                           ;*
                                           ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                           ;*F/FS OR GATES
                                           ;*

                                           ;;*************************************************************
002614  000004                             TST4:   SCOPE
002616  012737  000100  001160                     MOV     #100,$TIMES        ;;DO 100 ITERATIONS

002624  005077  176546                             CLR     @ASR               ;/CLEAR THE STATUS REGISTER.
002630  052777  020000  176540                     BIS     #BIT13,@ASR        ;/SET BIT 13.
002636  012737  020000  001124                     MOV     #BIT13,$GDDAT      ;/SET FOR ERROR TYPEOUT S/B.
002644  017737  176526  001126                     MOV     @ASR,$BDDAT        ;/READ THE STATUS REGISTER.
002652  023737  001124  001126                     CMP     $GDDAT,$BDDAT      ;/DID BIT 13 AND ONLY BIT 13 SET?
002660  001402                                     BEQ     1$                 ;/IF SO-LETS TRY CLEARING IT.

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

002662  104002                                     ERROR   2                  ;/ERROR CLOCK AS STATUS REGISTER
                                                                              ;/BIT 13 FAILED TO BIT SET.

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

002664  000412                                     BR      2$                 ;/BR TO END SUBTEST.

002666  042777  020000  176502             1$:     BIC     #BIT13,@ASR        ;/TRY CLEARING BIT 13.
002674  005037  001124                             CLR     $GDDAT             ;/CLEAR S/B FOR TYPEOUT IF ANY.
002700  017737  176472  001126                     MOV     @ASR,$BDDAT        ;/NOW READ IT BACK.
002706  001401                                     BEQ     2$                 ;/IF ZERO - NO ERROR!


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

002710  104002                                     ERROR   2                  ;/ERROR - CLOCK A STATUS REGISTER.
                                                                              ;/BIT 13 FAILED TO CLEAR.


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

002712                                     2$:
```

```
                                                ;/*
                                        ;;*****************************************************************
                                        ;*TEST 5        *TEST THAT CLOCK A STATUS REGISTER BIT 11 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*

                                        ;;*****************************************************************
002712  000004                          TST5:   SCOPE
002714  012737  000100  001160                  MOV     #100,$TIMES     ;;DO 100 ITERATIONS

002722  005077  176450                          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
002726  052777  004000  176442                  BIS     #BIT11,@ASR     ;/SET BIT 11.
002734  012737  004000  001124                  MOV     #BIT11,$GDDAT   ;/SET FOR ERROR TYPEOUT S/B.
002742  017737  176430  001126                  MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTER.
002750  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DID BIT 11 AND ONLY BIT 11 SET?
002756  001402                                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

002760  104002                                  ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                                        ;/BIT 11 FAILED TO BIT SET.

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

002762  000412                                  BR      2$              ;/BR TO END SUBTEST.

002764  042777  004000  176404          1$:     BIC     #BIT11,@ASR     ;/TRY CLEARING BIT 11.
002772  005037  001124                          CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
002776  017737  176374  001126                  MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003004  001401                                  BEQ     2$              ;/IF ZERO - NO ERROR!


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

003006  104002                                  ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                                        ;/BIT 11 FAILED TO CLEAR.


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

003010                                  2$:
```

```
                                        ;/*
                                ;;**********************************************************************
                                ;*TEST 6         *TEST THAT CLOCK A STATUS REGISTER BIT 6 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*

                                ;;**********************************************************************
003010  000004                  TST6:   SCOPE
003012  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS

003020  005077  176352          1$:     CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003024  052777  000100  176344          BIS     #BIT6,@ASR      ;/SET BIT 6.
003032  012737  000100  001124          MOV     #BIT6,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003040  017737  176332  001126          MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTER.
003046  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 6 AND ONLY BIT 6 SET?
003054  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003056  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                                ;/BIT 6 FAILED TO BIT SET.

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

003060  000412                          BR      2$              ;/BR TO END SUBTEST.

003062  042777  000100  176306  1$:     BIC     #BIT6,@ASR      ;/TRY CLEARING BIT 6.
003070  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003074  017737  176276  001126          MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003102  001401                          BEQ     2$              ;/IF ZERO - NO ERROR!

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

003104  104002                          ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                                ;/BIT 6 FAILED TO CLEAR.

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

003106                          2$:
```

```
                                        ;/*
                                        ;;**********************************************************************
                                        ;*TEST 7         *TEST THAT CLOCK A STATUS REGISTER BIT 5 CAN BE SET AND CLEARED
                                        ;*
                                        ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                        ;*F/FS OR GATES
                                        ;*
                                        ;;**********************************************************************
003106  000004                  TST7:   SCOPE
003110  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS

003116  005077  176254                  CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003122  052777  000040  176246          BIS     #BIT5,@ASR      ;/SET BIT 5.
003130  012737  000040  001124          MOV     #BIT5,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003136  017737  176234  001126          MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTER.
003144  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID BIT 5 AND ONLY BIT 5 SET?
003152  001402                          BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003154  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                                ;/BIT 5 FAILED TO BIT SET.

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

003156  000412                          BR      2$              ;/BR TO END SUBTEST.

003160  042777  000040  176210  1$:     BIC     #BIT5,@ASR      ;/TRY CLEARING BIT 5.
003166  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003172  017737  176200  001126          MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003200  001401                          BEQ     2$              ;/IF ZERO - NO ERROR!

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

003202  104002                          ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                                ;/BIT 5 FAILED TO CLEAR.

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

003204                          2$:
```

```
                                                 ;/*
                                           ;;*************************************************************
                                           ;*TEST 10      *TEST THAT CLOCK A STATUS REGISTER BIT 4 CAN BE SET AND CLEARED
                                           ;*
                                           ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                           ;*F/FS OR GATES
                                           ;*

                                           ;;*************************************************************
003204  000004                             TST10:  SCOPE
003206  012737  000100  001160                     MOV     #100,$TIMES        ;;DO 100 ITERATIONS

003214  005077  176156                             CLR     @ASR               ;/CLEAR THE STATUS REGISTER.
003220  052777  000020  176150                     BIS     #BIT4,@ASR         ;/SET BIT 4.
003226  012737  000020  001124                     MOV     #BIT4,$GDDAT       ;/SET FOR ERROR TYPEOUT S/B.
003234  017737  176136  001126                     MOV     @ASR,$BDDAT        ;/READ THE STATUS REGISTER.
003242  023737  001124  001126                     CMP     $GDDAT,$BDDAT      ;/DID BIT 4 AND ONLY BIT 4 SET?
003250  001402                                     BEQ     1$                 ;/IF SO-LETS TRY CLEARING IT.

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

003252  104002                                     ERROR   2                  ;/ERROR CLOCK AS STATUS REGISTER
                                                                              ;/BIT 4 FAILED TO BIT SET.

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

003254  000412                                     BR      2$                 ;/BR TO END SUBTEST.

003256  042777  000020  176112             1$:     BIC     #BIT4,@ASR         ;/TRY CLEARING BIT 4.
003264  005037  001124                             CLR     $GDDAT             ;/CLEAR S/B FOR TYPEOUT IF ANY.
003270  017737  176102  001126                     MOV     @ASR,$BDDAT        ;/NOW READ IT BACK.
003276  001401                                     BEQ     2$                 ;/IF ZERO - NO ERROR!


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

003300  104002                                     ERROR   2                  ;/ERROR - CLOCK A STATUS REGISTER.
                                                                              ;/BIT 4 FAILED TO CLEAR.


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

003302                                     2$:
```

```
                                                  ;/*
                                          ;;*************************************************************
                                          ;*TEST 11        *TEST THAT CLOCK A STATUS REGISTER BIT 3 CAN BE SET AND CLEARED
                                          ;*
                                          ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                          ;*F/FS OR GATES
                                          ;*
                                          ;;*************************************************************
003302  000004                            TST11:  SCOPE
003304  012737  000100  001160                    MOV     #100,$TIMES      ;;DO 100 ITERATIONS

003312  005077  176060                            CLR     @ASR             ;/CLEAR THE STATUS REGISTER.
003316  052777  000010  176052                    BIS     #BIT3,@ASR       ;/SET BIT 3.
003324  012737  000010  001124                    MOV     #BIT3,$GDDAT     ;/SET FOR ERROR TYPEOUT S/B.
003332  017737  176040  001126                    MOV     @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
003340  023737  001124  001126                    CMP     $GDDAT,$BDDAT    ;/DID BIT 3 AND ONLY BIT 3 SET?
003346  001402                                    BEQ     1$               ;/IF SO-LETS TRY CLEARING IT.

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

003350  104002                                    ERROR   2                ;/ERROR CLOCK AS STATUS REGISTER
                                                                           ;/BIT 3 FAILED TO BIT SET.

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

003352  000412                                    BR      2$               ;/BR TO END SUBTEST.

003354  042777  000010  176014    1$:             BIC     #BIT3,@ASR       ;/TRY CLEARING BIT 3.
003362  005037  001124                            CLR     $GDDAT           ;/CLEAR S/B FOR TYPEOUT IF ANY.
003366  017737  176004  001126                    MOV     @ASR,$BDDAT      ;/NOW READ IT BACK.
003374  001401                                    BEQ     2$               ;/IF ZERO - NO ERROR!


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

003376  104002                                    ERROR   2                ;/ERROR - CLOCK A STATUS REGISTER.
                                                                           ;/BIT 3 FAILED TO CLEAR.


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

003400                                    2$:
```

```
                                        ;/*
                                ;;**********************************************************
                                ;*TEST 12        *TEST THAT CLOCK A STATUS REGISTER BIT 2 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*

                                ;;**********************************************************
003400  000004                  TST12:  SCOPE
003402  012737  000100  001160          MOV     #100,$TIMES     ,;DO 100 ITERATIONS

003410  005077  175762          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003414  052777  000004  175754  BIS     #BIT2,@ASR      ;/SET BIT 2.
003422  012737  000004  001124  MOV     #BIT2,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003430  017737  175742  001126  MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTER.
003436  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;/DID BIT 2 AND ONLY BIT 2 SET?
003444  001402                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003446  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                                ;/BIT 2 FAILED TO BIT SET.

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

003450  000412                          BR      2$              ;/BR TO END SUBTEST.

003452  042777  000004  175716  1$:     BIC     #BIT2,@ASR      ;/TRY CLEARING BIT 2.
003460  005037  001124          CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003464  017737  175706  001126  MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003472  001401                  BEQ     2$              ;/IF ZERO - NO ERROR!

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

003474  104002                          ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                                ;/BIT 2 FAILED TO CLEAR.

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

003476                          2$:
```

```
                                        ;/*
                                ;;**********************************************************
                                ;*TEST 13       *TEST THAT CLOCK A STATUS REGISTER BIT 1 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*

                                ;;**********************************************************
003476  000004                  TST13:  SCOPE
003500  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS

003506  005077  175664          CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003512  052777  000002  175656  BIS     #BIT1,@ASR      ;/SET BIT 1.
003520  012737  000002  001124  MOV     #BIT1,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003526  017737  175644  001126  MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTER.
003534  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;/DID BIT 1 AND ONLY BIT 1 SET?
003542  001402                  BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

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

003544  104002                          ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                                ;/BIT 1 FAILED TO BIT SET.

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

003546  000412                          BR      2$              ;/BR TO END SUBTEST.

003550  042777  000002  175620  1$:     BIC     #BIT1,@ASR      ;/TRY CLEARING BIT 1.
003556  005037  001124                  CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003562  017737  175610  001126          MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003570  001401                  BEQ     2$              ;/IF ZERO - NO ERROR!

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

003572  104002                          ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                                ;/BIT 1 FAILED TO CLEAR.

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

003574                          2$:
```

```
                                        ;/*
                                ;;**************************************************************
                                ;*TEST 14        *TEST THAT CLOCK A STATUS REGISTER BIT 0 CAN BE SET AND CLEARED
                                ;*
                                ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
                                ;*F/FS OR GATES
                                ;*
                                ;;**************************************************************
003574  000004                TST14:  SCOPE
003576  012737  000100  001160        MOV     #100,$TIMES     ;;DO 100 ITERATIONS

003604  005077  175566              CLR     @ASR            ;/CLEAR THE STATUS REGISTER.
003610  052777  000001  175560      BIS     #BIT0,@ASR      ;/SET BIT 0.
003616  012737  000001  001124      MOV     #BIT0,$GDDAT    ;/SET FOR ERROR TYPEOUT S/B.
003624  017737  175546  001126      MOV     @ASR,$BDDAT     ;/READ THE STATUS REGISTER.
003632  023737  001124  001125      CMP     $GDDAT,$BDDAT   ;/DID BIT 0 AND ONLY BIT 0 SET?
003640  001402                      BEQ     1$              ;/IF SO-LETS TRY CLEARING IT.

                                ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

003642  104002                      ERROR   2               ;/ERROR CLOCK AS STATUS REGISTER
                                                        ;/BIT 0 FAILED TO BIT SET.

                                ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

003644  000412                      BR      2$              ;/BR TO END SUBTEST.

003646  042777  000001  175522  1$:     BIC     #BIT0,@ASR      ;/TRY CLEARING BIT 0.
003654  005037  001124              CLR     $GDDAT          ;/CLEAR S/B FOR TYPEOUT IF ANY.
003660  017737  175512  001126      MOV     @ASR,$BDDAT     ;/NOW READ IT BACK.
003666  001401                      BEQ     2$              ;/IF ZERO - NO ERROR!


                                ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

003670  104002                      ERROR   2               ;/ERROR - CLOCK A STATUS REGISTER.
                                                        ;/BIT 0 FAILED TO CLEAR.


                                ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

003672                          2$:

                                        .RADIX  8


                                ;;**************************************************************
                                ;*TEST 15        *TEST THAT PATERN 125252 WILL SET AND CLEAR IN BUFFER REG.
                                ;;**************************************************************
003672  000004                TST15:  SCOPE

003674  005077  175500              CLR     @ABR            ;/CLEAR THE BUFFER REG.
003700  012737  125252  001124      MOV     #125252,$GDDAT  ;/RECORD PATTERN: 125252
```

```
1734  003706  013777  001124  175464          MOV     $GDDAT,@ABR     ;/SET PATTERN IN BUFFER REG.
1735  003714  017737  175460  001126          MOV     @ABR,$BDDAT     ;/READ THE BUFFER REG.
1736
1737  003722  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID THE PATTERN SET OK?
1738  003730  001402                          BEQ     1$              ;/YES-TRY CLEARING IT.
1739
1740

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

1743  003732  104003                          ERROR   3               ;/ERROR PATTERN 125252 FAILED TO
1744                                                                  ;/SET PROPERLY IN BUFFER REG.
1745

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

1748  003734  000412                          BR      2$              ;/GOTO SCOPE LOOP.
1749
1750  003736  042777  125252  175434  1$:     BIC     #125252,@ABR    ;/TRY CLEARING PATTERN.
1751  003744  005037  001124                  CLR     $GDDAT          ;/EXPECT ZERO BACK.
1752  003750  017737  175424  001126          MOV     @ABR,$BDDAT     ;/READ BUFFER REG.,WAS IT ZERO?
1753  003756  001401                          BEQ     2$              ;/YES-NEXT TEST.
1754
1755

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

1758  003760  104003                          ERROR   3               ;/BUFFER REG. COULD NOT BE LOADED
1759                                                                  ;/TO A ZERO.
1760

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

1763  003762                           2$:
1764
1765
1766                                    ;;=====================================================================
1767                                    ;*TEST 16       *TEST THAT PATERN 052525 WILL SET AND CLEAR IN BUFFER REG.
1768                                    ;;=====================================================================
1769  003762  000004                    TST16:  SCOPE
1770
1771  003764  005077  175410                  CLR     @ABR            ;/CLEAR THE BUFFER REG.
1772  003770  012737  052525  001124          MOV     #052525,$GDDAT  ;/RECORD PATTERN: 052525
1773  003776  013777  001124  175374          MOV     $GDDAT,@ABR     ;/SET PATTERN IN BUFFER REG.
1774  004004  017737  175370  001126          MOV     @ABR,$BDDAT     ;/READ THE BUFFER REG.
1775
1776  004012  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID THE PATTERN SET OK?
1777  004020  001402                          BEQ     1$              ;/YES-TRY CLEARING IT.
1778
1779

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

1782  004022  104003                          ERROR   3               ;/ERROR PATTERN 052525 FAILED TO
1783                                                                  ;/SET PROPERLY IN BUFFER REG.
1784

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

1787  004024  000412                          BR      2$              ;/GOTO SCOPE LOOP.
```

```
                                                1788
004026  042777  052525  175344  1$:     BIC     #052525,@ABR    ;/TRY CLEARING PATTERN.
004034  005037  001124                   CLR     $GDDAT          ;/EXPECT ZERO BACK.
004040  017737  175334  001126           MOV     @ABR,$BDDAT     ;/READ BUFFER REG.,WAS IT ZERO?
004046  001401                           BEQ     2$              ;/YES-NEXT TEST.
                                                1793
                                                1794
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
004050  104003                                  ERROR   3               ;/BUFFER REG. COULD NOT BE LOADED
                                                1798                    ;/TO A ZERO.
                                                1799
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
004052                                  2$:
                                                1803
                                                1804
                                                1805
                                                1806                    .SBTTL  *
                                                1807                    .SBTTL  * PHASE 2 ADVANCED BASIC LOGIC TESTS
                                                1808                    .SBTTL  *
                                                1809
                                                1810
                                        ;;**************************************************************
                                        ;*TEST 17       *TEST THE LOW BYTE OPERATION OF CLOCK'S STATUS REGISTER
                                                1813
                                        ;*
                                        ;*WE CAN SUCCESSFULLY WRITE EVERY BIT IN STATUS REG A
                                        ;*NOW LETS CHECK THE BYTE OPERATION OF THIS REGISTER.
                                        ;*
                                                1818
                                        ;;**************************************************************
004052  000004                        TST17:  SCOPE
004054  012737  000050  001160               MOV     #50,$TIMES      ;;DO 50 ITERATIONS
                                                1822
004062  005077  175310                       CLR     @ASR            ;MAKE SURE THE STATUS REGISTER IS CLEAR.
004066  112777  127677  175302               MOVB    #127677,@ASR    ;TRY WRITING ALL BITS IN THE
                                                1825                    ;STATUS REGISTER. LOGIC SHOULD PREVENT IT
                                                1826                    ;FROM BEING WRITTEN INTO BECAUSE
                                                1827                    ;WE ARE USING A DATOB INSTRUCTION.
                                                1828
004074  017777  175276  175024               MOV     @ASR,@$BDDAT    ;NOW EXAMINE THE
                                                1830                    ;STATUS REGISTER.
004102  013737  001126  001124               MOV     $BDDAT,$GDDAT   ;FIX $GDDAT FOR ERROR TYPEOUT IF
004110  105037  001125                       CLRB    $GDDAT+1        ;ANY RROR HAS OCCURRED, UPPER BYTE CLEARED.
                                                1833
004114  105737  001127                       TSTB    $BDDAT+1        ;ARE ANY BITS IN THE UPPER BYTE
                                                1835                    ;OF THE STATUS REGISTER SET?
004120  001401                           BEQ     1$              ;BRANCH NEXT TEST IF UPPER BYTE=0.
                                                1837
                                                1838
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
004122  104001                                  ERROR   1               ;ERROR - WROTE INTO UPPER BYTE OF
```

```
                                                        ;CLOCK'S STATUS WHEN
                                                        ;DOING A DATOB TO THE LOW BYTE.


                        ;;SSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SSSSSSSSSSSSSSSSSSSSSSSSSSSS

004124                  1$:

                        ;;**********************************************************************
                        ;*TEST 20        *TEST THE HIGH BYTE OPERATION OF A'S STATUS REGISTER

                        ;*
                        ;*WE CAN SUCCESSFULLY WRITE EVERY BIT IN STATUS REG A
                        ;*NOW LETS CHECK THE BYTE OPERATION OF THIS REGISTER.
                        ;*

                        ;;**********************************************************************
004124  000004          TST20:  SCOPE
004126  012737  000050  001160          MOV     #50,$TIMES      ;;DO 50 ITERATIONS

004134  005077  175236                  CLR     @ASR            ;CLEAR THE STATUS REGISTER.

004140  005237  001376                  INC     ASR             ;ADD #1 TO THE STATUS REGISTER'S ADDRESS
                                                                ;SO THAT WE WILL BE WRITING INTO
                                                                ;THE HIGH BYTE.

004144  112777  177313  175224          MOVB    #177313,@ASR    ;TRY WRITING ALL BITS IN THE STATUS
                                                                ;REGISTER. LOGIC SHOULD PREVENT THE LOW
                                                                ;BYTE OF THE STATUS REGISTER FROM
                                                                ;BEING WRITTEN INTO BECAUSE WE ARE USING
                                                                ;A DATOB INSTRUCTION WITH ADD SET.

004152  005337  001376                  DEC     ASR             ;FIX ADDRESS OF THE STATUS REGISTER ADDR.
                                                                ;SO WE CAN LOOK AT THE WHOLE WORD.
004156  017737  175214  001126          MOV     @ASR,$BDDAT     ;READ BACK WHAT THE STATUS REG. CONTAINS
004164  013737  001126  001124          MOV     $BDDAT,$GDDAT   ;FIX $GDDAT FOR ERROR TYPEOUT IF AN ERROR
004172  105037  001124                  CLRB    $GDDAT          ;OCCURRED, LOWER BYTE CLEARED.
004176  105737  001126                  TSTB    $BDDAT          ;IS LOWER BYTE CLEAR?
004202  001401                          BEQ     1$              ;BR IF YES TO NEXT SUBTEST.


                        ;;SSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SSSSSSSSSSSSSSSSSSSSSSSSSSSS

004204  104001                          ERROR   1               ;ERROR - WROTE INTO LOWER BYTE
                                                                ;OF CLOCKS STATUS REGISTER WHEN
                                                                ;DOING A DATOB TO THE HIGH BYTE.

                        ;;SSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SSSSSSSSSSSSSSSSSSSSSSSSSSSS

004206                  1$:

                        ;;**********************************************************************
                        ;*TEST 21        *TEST CLOCK'S COUNT REGISTER WITH 125252 PATTERN
                        ;;**********************************************************************
```

```
1896  004206  000004                    TST21:  SCOPE
1897  004210  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1898
1899  004216  005077  175154                    CLR     @ASR            ;SELECT MODE 0.
1900  004222  012777  125252  175150            MOV     #125252,@ABR    ;LOAD THE BUFFER REGISTER WITH
1901                                                                    ;PATTERN 125252. IT WILL BE
1902                                                                    ;TRANSFERRED TO THE COUNT REGISTER
1903                                                                    ;SINCE THIS IS MODE 0.
1904  004230  052777  000001  175140            BIS     #BIT0,@ASR      ;SET GO BIT(ALLOWS BUFFER-COUNT REG XFER).
1905
1906  004236  012737  125252  001124            MOV     #125252,$GDDAT  ;SET EXPECTED TO PATTERN IN CASE OF
1907                                                                    ;NEED OF ERROR TYPEOUT.
1908  004244  017746  175126                    MOV     @ASR,-(6)       ;/SAVE CSR
1909  004250  011637  001424                    MOV     (6),$TMP3       ;/GET CSR.
1910  004254  042737  177707  001424            BIC     #177707,$TMP3   ;/SAVE RATE BITS.
1911  004262  052737  004005  001424            BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
1912  004270  013777  001424  175100            MOV     $TMP3,@ASR              ;/LOAD CSR.
1913                                                                    ;/THIS MUST BE DONE IN
1914                                                                    ;/ORDER TO XFERR COUNTER
1915                                                                    ;/TO BUFFER ON ST2.
1916  004276  052777  001000  175072            BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
1917  004304  017737  175070  001126            MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
1918                                                                    ;/PREVIOUS COUNTER
1919  004312  012677  175060                    MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
1920  004316  005737  001126                    TST     $BDDAT          ;/RESTORE CSR
1921
1922  004322  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
1923                                                                    ;COME THROUGH?
1924  004330  001401                            BEQ     1$              ;BR IF YES TO NEXT TEST.
1925
1926
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

1929  004332  104005                            ERROR   5               ;DATA ERROR CLOCK - PATTERN "125252"
1930                                                                    ;FAILED TO TRANSFER PROPERLY BETWEEN
1931                                                                    ;BUFFER AND COUNT REGISTERS.
1932
1933
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

1936  004334                            1$:
1937
1938                                    ;;*****************************************************************
1939                                    ;*TEST 22        *TEST CLOCKS COUNTER REGISTER WITH 052525 PATTERN
1940                                    ;;*****************************************************************
1941  004334  000004                    ST22:   SCOPE
1942  004336  012737  000050  001160            MOV     #50,$TIMES      ;;DO 50 ITERATIONS
1943
1944  004344  005077  175026                    CLR     @ASR            ;SELECT MODE 0.
1945  004350  012777  052525  175022            MOV     #052525,@ABR    ;LOAD THE BUFFER REGISTER WITH
1946                                                                    ;PATTERN 052525. IT WILL BE
1947                                                                    ;TRANSFERRED TO THE COUNT REGISTER
1948                                                                    ;SINCE THIS IS MODE 0.
1949  004356  052777  000001  175012            BIS     #BIT0,@ASR      ;SET BO BIT(ALLOWS BUFFER-COUNT REG XFER).
```

```

004364  012737  052525  001124          MOV     #052525,$GDDAT  ;SET EXPECTED TO PATTERN IN CASE OF
                                                                ;NEED OF ERROR TYPEOUT.
004372  017746  175000                  MOV     @ASR,-(6)       ;/SAVE CSR
004376  011637  001424                  MOV     (6),$TMP3       ;/GET CSR.
004402  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
004410  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
004416  013777  001424  174752          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                ;/THIS MUST BE DONE IN
                                                                ;/ORDER TO XFERR COUNTER
                                                                ;/TO BUFFER ON ST2.
004424  052777  001000  174744          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
004432  017737  174742  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
                                                                ;/PREVIOUS COUNTER
004440  012677  174732                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
004444  005737  001126                  TST     $BDDAT          ;/RESTORE CSR

004450  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID ALL THE BITS AND NO OTHER BITS
                                                                ;COME THROUGH?
004456  001401                          BEQ     1$              ;BR IF YES TO NEXT TEST.


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

004460  104005                          ERROR   5               ;DATA ERROR CLOCK - PATTERN "052525"
                                                                ;FAILED TO TRANSFER PROPERLY BETWEEN
                                                                ;BUFFER AND COUNT REGISTERS.



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

004462                          1$:



                                ;;**************************************************************
                                ;*TEST 23       *TEST THAT INIT CLEARS STATUS REGISTER
                                ;*
                                ;*TESTING OF THE INIT LOGIC AS RECEIVED FROM THE QBUS AND BUFFERED
                                ;*TO STATUS REGISTER F/FS.
                                ;*
                                ;;**************************************************************
004462  000004                  TST23:  SCOPE
004464  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

004472  005037  001124                  CLR     $GDDAT          ;EXPECTED DATA IS ZERO.
004476  012777  177777  174672          MOV     #177777,@ASR    ;SET ALL BITS IN THE STATUS REG.

004504  000005                          RESET                   ;SYSTEM INITIALIZE.

004506  017737  174664  001126          MOV     @ASR,$BDDAT     ;READ THE STATUS REG., ALL BITS SHOULD
                                                                ;HAVE BEEN CLEARED BY INIT.
004514  001402                          BEQ     1$              ;BR IF YES TO NEXT TEST.
```

F04


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

004516  104002                                  ERROR   2               ;ERROR - SYSTEM INIT FAILED TO CLEAR
                                                                        ;STATUS REGISTER CLOCK A.


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

004520  000413                                  BR      TST24           ;;
004522                                  1$:
004522  005737  001440                          TST     EXS             ;TEST EXTERNAL SIGS?
004526  001410                                  BEQ     TST24           ;;
004530  052777  016000  174660                  BIS     #BIT11!BIT12!BIT10,@DR2 ;ENABLE ST1,ST2 TO LATCH.
004536  032777  000006  174650                  BIT     #6,@DR          ;ST1,ST2, OVERFLOW SET?
004544  001401                                  BEQ     TST24           ;;

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

004546  104006                                  ERROR   6               ;INIT FAILED TO CLEAR
                                                                        ;ST1,ST2, AND/OR OVERFLOW

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

                                        ;;******************************************************************
                                        ;*TEST 24        *TEST THAT INIT CLEARS BUFFER REGISTER
                                        ;*
                                        ;*THIS TEST IS DESIGNED TO SEE IF "INIT H"
                                        ;*CLEARS THE BUFFER REGISTER. WE ALREADY
                                        ;*KNOW IT CLEARS THE STATUS REG.
                                        ;*
                                        ;;******************************************************************
004550  000004                          TST24:  SCOPE
004552  012737  000005  001160                  MOV     #5,$TIMES       ;;DO 5 ITERATIONS

004560  005037  001124                          CLR     $GDDAT          ;CLEAR EXPECTED DATA.
004564  012777  177777  174606                  MOV     #177777,@ABR    ;SET ALL BITS IN THE BUFFER REGISTER.

004572  000005                                  RESET                   ;ISSUE SYSTEM INITIALIZE.

004574  017737  174600  001126                  MOV     @ABR,$BDDAT     ;READ THE BUFFER REGISTER, ALL BITS
                                                                        ;SHOULD HAVE BEEN CLEARED BY INIT.
004602  001401                                  BEQ     1$              ;BR IF YES TO NEXT SUBTEST.


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

004604  104003                                  ERROR   3               ;ERROR - SYSTEM INIT FAILED
                                                                        ;TO CLEAR BUFFER REGISTER A.


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

```
2058  004606                          1$:
2059  004606  005737  001440                  TST     EXS             ;TEST EXTERNAL SIGS?
2060  004612  001403                          BEQ     TST25           ;;
2061  004614  052777  016000  174574          BIS     #BIT11!BIT12!BIT10,@DR2 ;ENABLE THEM
2062
2063                                  ;;*****************************************************************
2064                                  ;*TEST 25       *TEST THE SETTING OF MAINTENANCE ST2 IN CLOCK BIT 15 TO SET
2065                                  ;;*****************************************************************
2066  004622  000004                  TST25:  SCOPE
2067
2068  004624  012777  000001  174544          MOV     #1,@ASR         ;SET THE GO BIT(ENABLES ST2 TO SET FLG).
2069  004632  052777  001000  174536          BIS     #BIT9,@ASR      ;SET MAINTENANCE ST2.
2070
2071  004640  005777  174532                  TST     @ASR            ;DID BIT15 (ST2 FLAG) SET?
2072  004644  100402                          BMI     1$              ;BR IF YES - NEXT TEST
2073
2074
                                    ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

2077  004646  104001                          ERROR   1               ;ERROR - MAINTENANCE ST2 (BIT9)
2078                                                                  ;DID NOT SET BIT15 (ST2 FLAG).
2079
2080
                                    ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

2083  004650  000410                          BR      TST26           ;;
2084  004652  005737  001440          1$:     TST     EXS             ;TEST EXTERNAL SIGNALS?
2085  004656  001405                          BEQ     TST26           ;;
2086  004660  032777  000004  174526          BIT     #BIT2,@DR       ;DID EXTERNAL ST2 GET SET?
2087  004666  001001                          BNE     TST26           ;;
2088
                                    ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

2091  004670  104006                          ERROR   6               ;ST2 OUT NOT DETECTED
2092                                                                  ;BY TESTOR
2093
                                    ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

2096
2097
2098                                  ;;*****************************************************************
2099                                  ;*TEST 26       *TEST THAT BIT00 IN CLOCK STATUS REG. WILL SET WHEN BIT13 AND MAIN. ST2
2100                                  ;;*****************************************************************
2101  004672  000004                  TST26:  SCOPE
2102
2103  004674  012777  020000  174474          MOV     #BIT13,@ASR     ;SET "ST2 ENB COUNTER" IN CLK  STATUS REG.
2104  004702  052777  001000  174466          BIS     #BIT9,@ASR      ;GENERATE A MAINTENANCE ST2.
2105  004710  032777  000001  174460          BIT     #BIT00,@ASR     ;DID BIT00 (GO) SET?
2106  004716  001001                          BNE     1$              ;BR IF YES - NEXT TEST.
2107
2108
                                    ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

2111  004720  104001                          ERROR   1               ;ERROR - BIT00 OF CLOCK'S STATUS REGISTER
```

```
                                                                ;FAILED TO SET WHEN BIT13 WAS SET
                                                                ;AND A MAINTENANCE ST2 GENERATED.

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

004722  005077  174450                  1$:     CLR     @ASR            ;LEAVE SUBTEST WITH CLOCK CLEAR.

                                                .SBTTL  *
                                                .SBTTL  *PHASE 3 COUNT TESTS
                                                .SBTTL  *

                                        ;;*************************************************************
                                        ;*TEST 27       *TEST TO SEE IF THE COUNTER WILL INCREMENT
                                        ;;*************************************************************
004726  000004                          TST27:  SCOPE

004730  005077  174442                          CLR     @ASR            ;CLEAR THE CSR.
004734  005077  174440                          CLR     @ABR            ;CLEAR THE BUFFER
004740  052777  000061  174430                  BIS     #BIT5!BIT4!BIT0,@ASR ;SET RATE:ST1, GO.
004746  052777  000400  174422                  BIS     #BIT8,@ASR      ;GENERATE A MAINTENANCE ST1.
                                                                        ;DID THE CLOCK COUNT?
004754  017746  174416                          MOV     @ASR,-(6)       ;/SAVE CSR
004760  011637  001424                          MOV     (6),$TMP3       ;/GET CSR.
004764  042737  177707  001424                  BIC     #177707,$TMP3   ;/SAVE RATE BITS.
004772  052737  004005  001424                  BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
005000  013777  001424  174370                  MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                        ;/THIS MUST BE DONE IN
                                                                        ;/ORDER TO XFERR COUNTER
                                                                        ;/TO BUFFER ON ST2.
005006  052777  001000  174362                  BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
005014  017737  174360  001126                  MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
                                                                        ;/PREVIOUS COUNTER
005022  012677  174350                          MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
005026  005737  001126                          TST     $BDDAT          ;/RESTORE CSR
005032  001001                                  BNE     1$              ;YES, NEXT TEST.

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

005034  104006                                  ERROR   6               ;CLOCK FAILED TO INCREMENT.

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

005036                                  1$:

                                        ;;*************************************************************
                                        ;*TEST 30       *SEE IF CLOCK WILL COUNT UP FROM A ZERO BASE. RATE:ST1
                                        ;*
                                        ;*      NOTE:   IN THIS TEST, LOOP ON ERROR WILL CAUSE A LOOP
                                        ;*              ON THE FAILING COUT PATTERN;
                                        ;*              WHILE LOOP ON TEST WILL START THE TEST
                                        ;*              AND THE CLOCK FROM ZERO TO THE FAILING COUNT.
                                        ;*
```

```
2166                                            ;;**************************************************************
2167  005036  000004                    TST30:  SCOPE
2168  005040  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2169
2170  005046  005077  174324                    CLR     @ASR            ;CLEAR THE CSR.
2171  005052  005077  174322                    CLR     @ABR            ;CLEAR THE BUFFER REG
2172  005056  012737  000000  001124            MOV     #0,$GDDAT       ;CLEAR EXPECTED.
2173  005064  012737  005226  001110            MOV     #2$,$LPERR
2174
2175  005072  052777  000061  174276  1$:       BIS     #BIT5!BIT4!BIT0,@ASR ;START CLOCK, RATE:ST1.
2176
2177  005100  052777  000400  174270            BIS     #BIT8,@ASR      ;GENERATE A MAINTENANCE ST1
2178                                                                    ;CLOCK SHOULD COUNT ONCE.
2179  005106  005737  001440                    TST     EXS             ;TEST EXTERNAL SIGNALS?
2180  005112  001406                            BEQ     10$             ;NO
2181  005114  032777  000002  174272            BIT     #BIT1,@DR       ;YES - DID ST1 GET SET?
2182  005122  001002                            BNE     10$             ;YES
2183
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2186  005124  104006                            ERROR   6               ;ST1 OUT NOT DETECTED
2187                                                                    ;BY TESTOR
2188
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2191  005126  000447                            BR      TST31           ;;
219   005130                            10$:
219   005130  017746  174242                    MOV     @ASR,-(6)       ;/SAVE CSR
2194  005134  011637  001424                    MOV     (6),$TMP3       ;/GET CSR.
2195  005140  042737  177707  001424            BIC     #177707,$TMP3   ;/SAVE RATE BITS.
2196  005146  052737  004005  001424            BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2197  005154  013777  001424  174214            MOV     $TMP3,@ASR              ;/LOAD CSR.
2198                                                                    ;/THIS MUST BE DONE IN
2199                                                                    ;/ORDER TO XFERR COUNTER
2200                                                                    ;/TO BUFFER ON ST2.
2201  005162  052777  001000  174206            BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
2202  005170  017737  174204  001126            MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
2203                                                                    ;/PREVIOUS COUNTER
2204  005176  012677  174174                    MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
2205  005202  005737  001126                    TST     $BDDAT          ;/RESTORE CSR
2206  005206  005237  001124                    INC     $GDDAT          ;COUNT=OLD COUNT+1
2207  005212  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COUNT READ=COUNT EXP'ED?
2208  005220  001402                            BEQ     2$              ;YES - SEE IF WE'RE THROUGH.
2209
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2212  005222  104011                            ERROR   11              ;CLOCK FAILED TO COUNT UP PROPERLY.
2213
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2216  005224  000410                            BR      3$              ;GOTO SCOPE LOOP.
2217
2218  005226  005077  174144            2$:     CLR     @ASR
2219  005232  013777  001124  174140            MOV     $GDDAT,@ABR
```

```
                                        TST     $GDDAT          ;ALL DONE?
                                        BNE     1$              ;NO DO NEXT INCREMENT.
                                3$:
                                ;;**************************************************************
                                ;*TEST 31       *TEST THAT OVERFLOW (CSR BIT07) WILL SET ON OVERFLOW
                                ;;**************************************************************
                                TST31:  SCOPE
                                        TST     EXS             ;TESTOR MODE ENABLED??
                                        BEQ     2$              ;NO-THEN SKIP NEXT SECTION OF CODE.
                                        CLR     @#170500        ;CLEAR TESTOR A/D
                                        TST     @#170502        ;DUMB READ OF A/D BUFFER.
                                        BIS     #BIT5,@#170500  ;ENABLE EXTRENAL START OF A/D.
                                        MOV     #8.,R0          ;SET TIME OUT NUMBER.
                                2$:
                                        CLR     @ASR            ;CLEAR THE CSR
                                        MOV     #-1,@ABR        ;SET PRESET BUFFER TO ALL ONES.

                                        BIS     #BIT5!BIT4!BIT0,@ASR ;START CLOCK, RATE ST1.

                                        BIS     #BIT8,@ASR      ;COUNT CLOCK ONCE, OVERFLOW
                                                                ;SHOULD OCCUR.
                                        TSTB    @ASR            ;DID OVERFLOW SET?
                                        BMI     1$              ;YES - THEN NEXT TEST


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

                                        ERROR   6               ;ERROR - OVERFLOW, CSR BIT0?
                                                                ;FAILED TO SET ON OVERFLOW
                                        BR      TST32           ;;
                                1$:     TST     EXS             ;TEST EXTERNAL SIGNALS?
                                        BEQ     TST32           ;;
                                        TSTB    @#170500        ;IF  OVERFLOW GOT OUT,IT GAVE A/D START,
                                                                ;WE'RE LOOKING FOR A/D DONE-DID IT GET SET?
                                        BMI     TST32           ;;
                                        DEC     R0              ;DID WE ALLOW ENOUGH TIME??
                                        BNE     1$              ;NO-THEN WAIT.

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

                                        ERROR   6               ;OVERFLOW OUT NOT DETECTED
                                                                ;BY TESTOR

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


                                ;;**************************************************************
                                ;*TEST 32       *TEST THAT OVERFLOW WILL CLEAR THE GO BIT
                                ;;**************************************************************
                                TST32:  SCOPE
```

```

005364  005077  174006                  CLR     @ASR            ;CLEAR THE CSR.

005370  012777  177777  174002          MOV     #-1,@ABR        ;PRESET CLOCK TO -1.

005376  052777  000061  173772          BIS     #BIT5!BIT4!BIT0,@ASR ;START CLOCK, RATE:ST1

005404  052777  000400  173764          BIS     #BIT8,@ASR      ;COUNT ONCE, OVERFLOW
                                                                ;SHOULD OCCUR CLEARING
                                                                ;ENABLE (CSR BIT00)

005412  032777  000001  173756          BIT     #BIT0,@ASR      ;DID THE ENABLE CLEAR?
005420  001401                          BEQ     1$              ;YES - NEXT TEST.


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

005422  104006                          ERROR   6               ;ERROR - OVERFLOW FAILED
                                                                ;TO CLEAR ENABLE (CSR BIT00)

005424                          1$:


                                ;;*************************************************************
                                ;*TEST 33       *TEST THAT GO BIT DOES NOT CLEAR ON OVERFLOW, IF MODE 1
                                ;;*************************************************************
005424  000004                  TST33:  SCOPE

005426  005077  173744                  CLR     @ASR            ;CLEAR THE CSR.
005432  012777  177777  173740          MOV     #-1,@ABR        ;PRESET BUFFER=ONE COUNT FROM OVERFLOW.
005440  052777  000063  173730          BIS     #63,@ASR        ;MODE 1, RATE:ST1, GO.

005446  052777  000400  173722          BIS     #BIT8,@ASR      ;GENERATE MAINTENANCE ST1.

005454  032777  000001  173714          BIT     #BIT0,@ASR      ;DID ENABLE (GO BIT) CLEAR?
005462  001001                          BNE     1$              ;NO (GOOD) NEXT TEST.

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

005464  104006                          ERROR   6               ;GO BIT CLEARED ON OVERFLOW
                                                                ;WHEN MODE 1 WAS SELECTED

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

005466  005077  173704          1$:     CLR     @ASR            ;CLEAR THE CLOCK.





                                ;;*************************************************************
                                ;*TEST 34       *TEST THE ABILITY OF CLOCK TO COUNT AT 1MHZ RATE

                                ;*
                                ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
```

```
                                        ;*TO COUNT AT 1MHZ RATE.
                                        ;*
                                        ;;*****************************************************************
005472  000004                  $T34:   SCOPE
005474  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

005502  005077  173670          CLR     @ASR            ;/CLEAR CLOCK
005506  005077  173666          CLR     @ABR            ;/CLEAR PRESET BUFFER
005512  012777  000011  173656  MOV     #BIT0!10,@ASR   ;/START CLOCK, MODE0, RATE:1MHZ
005520  005000                  CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY

005522  005200          1$:     INC     R0              ;/WILL AMOUNT TO APPROXIMATELY
005524  001376                  BNE     1$              ;/369 MS.

005526  017746  173644          MOV     @ASR,-(6)       ;/SAVE CSR
005532  011637  001424          MOV     (6),$TMP3       ;/GET CSR.
005536  042737  177707  001424  BIC     #177707,$TMP3   ;/SAVE RATE BITS.
005544  052737  004005  001424  BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
005552  013777  001424  173616  MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                        ;/THIS MUST BE DONE IN
                                                        ;/ORDER TO XFERR COUNTER
                                                        ;/TO BUFFER ON ST2.
005560  052777  001000  173610  BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
005566  017737  173606  001126  MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
                                                        ;/PREVIOUS COUNTER
005574  012677  173576          MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
005600  005737  001126          TST     $BDDAT          ;/RESTORE CSR
005604  001004                  BNE     2$              ;/YES - NEXT TEST.
005606  105766  177776          TSTB    -2(6)           ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
                                                        ;/NOTE: CSR HAD BEEN PUT ON STACK.
005612  100401                  BMI     2$              ;/NEXT TEST IF OVERFLOW.


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

005614  104006                  ERROR   6               ;/CLOCK FAILED TO COUNT AT
                                                        ;/RATE:1MHZ


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


005616  005077  173554  2$:     CLR     @ASR            ;/CLEAR THE CLOCK.



                                ;;*****************************************************************
                                ;*TEST 35        *TEST THE ABILITY OF CLOCK TO COUNT AT 100KHZ RATE

                                ;*
                                ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
                                ;*TO COUNT AT 100KHZ RATE.
                                ;*
                                ;;*****************************************************************
```

```
2382  005622  000004                    TST35:  SCOPE
2383  005624  012737  000005  001160            MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2384
2385
2386  005632  005077  173540                    CLR     @ASR            ;/CLEAR CLOCK
2387  005636  005077  173536                    CLR     @ABR            ;/CLEAR PRESET BUFFER
2388  005642  012777  000021  173526            MOV     #BIT0!20,@ASR   ;/START CLOCK, MODE0, RATE:100KHZ
2389  005650  005000                            CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
2390
2391  005652  005200                    1$:     INC     R0              ;/WILL AMOUNT TO APPROXIMATELY
2392  005654  001376                            BNE     1$              ;/369 MS.
2393
2394  005656  017746  173514                    MOV     @ASR,-(6)       ;/SAVE CSR
2395  005662  011637  001424                    MOV     (6),$TMP3       ;/GET CSR.
2396  005666  042737  177707  001424            BIC     #177707,$TMP3   ;/SAVE RATE BITS.
2397  005674  052737  004005  001424            BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2398  005702  013777  001424  173466            MOV     $TMP3,@ASR              ;/LOAD CSR.
2399                                                                    ;/THIS MUST BE DONE IN
2400                                                                    ;/ORDER TO XFERR COUNTER
2401                                                                    ;/TO BUFFER ON ST2.
2402  005710  052777  001000  173460            BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
2403  005716  017737  173456  001126            MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
2404                                                                    ;/PREVIOUS COUNTER
2405  005724  012677  173446                    MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
2406  005730  005737  001126                    TST     $BDDAT          ;/RESTORE CSR
2407  005734  001004                            BNE     2$              ;/YES - NEXT TEST.
2408  005736  105766  177776                    TSTB    -2(6)           ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
2409                                                                    ;/NOTE: CSR HAD BEEN PUT ON STACK.
2410  005742  100401                            BMI     2$              ;/NEXT TEST IF OVERFLOW.
2411
2412
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2415  005744  104006                            ERROR   6               ;/CLOCK FAILED TO COUNT AT
2416                                                                    ;/RATE:100KHZ
2417
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2420
2421  005746  005077  173424            2$:     CLR     @ASR            ;/CLEAR THE CLOCK.
2422
2423
2424
2425                                    ;;*****************************************************************
2426                                    ;*TEST 36        *TEST THE ABILITY OF CLOCK TO COUNT AT 10KHZ RATE
2427
2428                                    ;*
2429                                    ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
2430                                    ;*TO COUNT AT 10KHZ RATE.
2431                                    ;*
2432                                    ;*****************************************************************
2433  005752  000004                    TST36:  SCOPE
2434  005754  012737  000005  001160            MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2435
```

```
2436
2437  005762  005077  173410                CLR     @ASR            ;/CLEAR CLOCK
2438  005766  005077  173406                CLR     @ABR            ;/CLEAR PRESET BUFFER
2439  005772  012777  000031  173376        MOV     #BIT0!30,@ASR   ;/START CLOCK, MODE0, RATE:10KHZ
2440  006000  005000                        CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
2441
2442  006002  005200                1$:     INC     R0              ;/WILL AMOUNT TO APPROXIMATELY
2443  006004  001376                        BNE     1$              ;/369 MS.
2444
2445  006006  017746  173364                MOV     @ASR,-(6)       ;/SAVE CSR
2446  006012  011637  001424                MOV     (6),$TMP3       ;/GET CSR.
2447  006016  042737  177707  001424        BIC     #177707,$TMP3   ;/SAVE RATE BITS.
2448  006024  052737  004005  001424        BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2449  006032  013777  001424  173336        MOV     $TMP3,@ASR              ;/LOAD CSR.
2450                                                                ;/THIS MUST BE DONE IN
2451                                                                ;/ORDER TO XFERR COUNTER
2452                                                                ;/TO BUFFER ON ST2.
2453  006040  052777  001000  173330        BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
2454  006046  017737  173326  001126        MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
2455                                                                ;/PREVIOUS COUNTER
2456  006054  012677  173316                MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
2457  006060  005737  001126                TST     $BDDAT          ;/RESTORE CSR
2458  006064  001004                        BNE     2$              ;/YES - NEXT TEST.
2459  006066  105766  177776                TSTB    -2(6)           ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
2460                                                                ;/NOTE: CSR HAD BEEN PUT ON STACK.
2461  006072  100401                        BMI     2$              ;/NEXT TEST IF OVERFLOW.
2462
2463
                                      ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2466  006074  104006                        ERROR   6               ;/CLOCK FAILED TO COUNT AT
2467                                                                ;/RATE:10KHZ
2468
                                      ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2471
2472  006076  005077  173274          2$:     CLR     @ASR            ;/CLEAR THE CLOCK.
2473
2474
2475
2476                                  ;;**************************************************** ************
2477                                  ;*TEST 37        *TEST THE ABILITY OF CLOCK TO COUNT AT 1KHZ RATE
2478
2479                                  ;*
2480                                  ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
2481                                  ;*TO COUNT AT 1KHZ RATE.
2482                                  ;*
2483                                  ;;*************************************************************
2484  006102  000004                  T$T37:  SCOPE
2485  006104  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2486
2487
2488  006112  005077  173260                  CLR     @ASR            ;/CLEAR CLOCK
2489  006116  005077  173256                  CLR     @ABR            ;/CLEAR PRESET BUFFER
```

```
006122  012777  000041  173246          MOV     #BIT0!40,@ASR       ;/START CLOCK, MODE0, RATE:1KHZ
006130  005000                          CLR     R0                  ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY

006132  005200                  1$:     INC     R0                  ;/WILL AMOUNT TO APPROXIMATELY
006134  001376                          BNE     1$                  ;/369 MS.

006136  017746  173234                  MOV     @ASR,-(6)           ;/SAVE CSR
006142  011637  001424                  MOV     (6),$TMP3           ;/GET CSR.
006146  042737  177707  001424          BIC     #177707,$TMP3       ;/SAVE RATE BITS.
006154  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
006162  013777  001424  173206          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                    ;/THIS MUST BE DONE IN
                                                                    ;/ORDER TO XFERR COUNTER
                                                                    ;/TO BUFFER ON ST2.
006170  052777  001000  173200          BIS     #BIT9,@ASR          ;/GENERATE ON ST2 PULSE
006176  017737  173176  001126          MOV     @ABR,$BDDAT         ;/READ THE PRESET BUFFER,
                                                                    ;/PREVIOUS COUNTER
006204  012677  173166                  MOV     (6)+,@ASR           ;/CONTENTS ARE IN $BDDAT.
006210  005737  001126                  TST     $BDDAT              ;/RESTORE CSR
006214  001004                          BNE     2$                  ;/YES - NEXT TEST.
006216  105766  177776                  TSTB    -2(6)               ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
                                                                    ;/NOTE: CSR HAD BEEN PUT ON STACK.
006222  100401                          BMI     2$                  ;/NEXT TEST IF OVERFLOW.


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

006224  104006                          ERROR   6                   ;/CLOCK FAILED TO COUNT AT
                                                                    ;/RATE:1KHZ

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


006226  005077  173144          2$:     CLR     @ASR                ;/CLEAR THE CLOCK.



                                ;;*************************************************************
                                ;*TEST 40        *TEST THE ABILITY OF CLOCK TO COUNT AT 100HZ RATE

                                ;*
                                ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
                                ;*TO COUNT AT 100HZ RATE.
                                ;*
                                ;;*************************************************************
006232  000004                  TST40:  SCOPE
006234  012737  000005  001160          MOV     #5,$TIMES           ;;DO 5 ITERATIONS


006242  005077  173130                  CLR     @ASR                ;/CLEAR CLOCK
006246  005077  173126                  CLR     @ABR                ;/CLEAR PRESET BUFFER
006252  012777  000051  173116          MOV     #BIT0!50,@ASR       ;/START CLOCK, MODE0, RATE:100HZ
006260  005000                          CLR     R0                  ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
```

```
2544  006262  005200                  1$:     INC     R0              ;/WILL AMOUNT TO APPROXIMATELY
2545  006264  001376                          BNE     1$              ;/369 MS.
2546
2547  006266  017746  173104                  MOV     @ASR,-(6)       ;/SAVE CSR
2548  006272  011637  001424                  MOV     (6),$TMP3       ;/GET CSR.
2549  006276  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
2550  006304  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2551  006312  013777  001424  173056          MOV     $TMP3,@ASR              ;/LOAD CSR.
2552                                                                  ;/THIS MUST BE DONE IN
2553                                                                  ;/ORDER TO XFERR COUNTER
2554                                                                  ;/TO BUFFER ON ST2.
2555  005320  052777  001000  173050          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
2556  006326  017737  173046  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
2557                                                                  ;/PREVIOUS COUNTER
2558  006334  012677  173036                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
2559  006340  005737  001126                  TST     $BDDAT          ;/RESTORE CSR
2560  006344  001004                          BNE     2$              ;/YES - NEXT TEST.
2561  006346  105766  177776                  TSTB    -2(6)           ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
2562                                                                  ;/NOTE: CSR HAD BEEN PUT ON STACK.
2563  006352  100401                          BMI     2$              ;/NEXT TEST IF OVERFLOW.
2564
2565                                  /
                                      ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2568  006354  104006                          ERROR   6               ;/CLOCK FAILED TO COUNT AT
2569                                                                  ;/RATE:100HZ
2570
                                      ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2573
2574  006356  005077  173014          2$:     CLR     @ASR            ;/CLEAR THE CLOCK.
2575
2576
2577
2578                                      ;;*****************************************************************
2579                                      ;*TEST 41        *TEST THE ABILITY OF CLOCK TO COUNT AT LINEFREQ RATE
2580
2581                                      ;*
2582                                      ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
2583                                      ;*TO COUNT AT LINEFREQ RATE.
2584                                      ;*
2585                                      ;*****************************************************************
2586  006362  000004                  TST41:  SCOPE
2587  006364  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2588
2589  006372  032777  010000  172540          BIT     #BIT12,@SWR     ;SHALL WE TEST
2590                                                                  ;LINE FREQ?
2591                                                                  ;SW12=1
2592  006400  001450                          BEQ     TST42           ;;
2593
2594  006402  005077  172770                  CLR     @ASR            ;/CLEAR CLOCK
2595  006406  005077  172766                  CLR     @ABR            ;/CLEAR PRESET BUFFER
2596  006412  012777  000071  172756          MOV     #BIT0!70,@ASR   ;/START CLOCK, MODE0, RATE:LINEFREQ
2597  006420  005000                          CLR     R0              ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
```

```

006422  005200                          1$:     INC     R0              ;/WILL AMOUNT TO APPROXIMATELY
006424  001376                                  BNE     1$              ;/369 MS.

006426  017746  172744                          MOV     @ASR,-(6)       ;/SAVE CSR
006432  011637  001424                          MOV     (6),$TMP3       ;/GET CSR.
006436  042737  177707  001424                  BIC     #177707,$TMP3   ;/SAVE RATE BITS.
006444  052737  004005  001424                  BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
006452  013777  001424  172716                  MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                        ;/THIS MUST BE DONE IN
                                                                        ;/ORDER TO XFERR COUNTER
                                                                        ;/TO BUFFER ON ST2.
006460  052777  001000  172710                  BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
006466  017737  172706  001126                  MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER.
                                                                        ;/PREVIOUS COUNTER
006474  012677  172676                          MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
006500  005737  001126                          TST     $BDDAT          ;/RESTORE CSR
006504  001004                                  BNE     2$              ;/YES - NEXT TEST.
006506  105766  177776                          TSTB    -2(6)           ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
                                                                        ;/NOTE: CSR HAD BEEN PUT ON STACK.
006512  100401                                  BMI     2$              ;/NEXT TEST IF OVERFLOW.


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

006514  104006                                  ERROR   6               ;/CLOCK FAILED TO COUNT AT
                                                                        ;/RATE:LINEFREQ

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


006516  005077  172654                  2$:     CLR     @ASR            ;/CLEAR THE CLOCK.




                                        ;;***********************************************************************
                                        ;*TEST 42        *TEST THAT COUNTER DOESN'T COUNT WHEN "SLAVE IN" RATE IS SELECTED
                                        ;;***********************************************************************
006522  000004                          TST42:  SCOPE
006524  012737  000020  001160                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS

006532  005077  172640                          CLR     @ASR            ;CLEAR CSR.
006536  005077  172636                          CLR     @ABR            ;CLEAR PRESET BUFFER.

006542  052777  000001  172626                  BIS     #BIT0,@ASR      ;SET GO BIT, RATE:"SLAVE IN"

006550  052777  000400  172620                  BIS     #BIT8,@ASR      ;GENERATE A MAINTENANCE ST1.
006556  005000                                  CLR     R0

006560  005200                          1$:     INC     R0              ;NOW DO A SHORT DELAY.
006562  001376                                  BNE     1$

006564  017746  172606                          MOV     @ASR,-(6)       ;/SAVE CSR
006570  011637  001424                          MOV     (6),$TMP3       ;/GET CSR.
```

```
2652  006574  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
2653  006602  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2654  006610  013777  001424  172560          MOV     $TMP3,@ASR              ;/LOAD CSR.
2655                                                                          ;/THIS MUST BE DONE IN
2656                                                                          ;/ORDER TO XFERR COUNTER
2657                                                                          ;/TO BUFFER ON ST2.
2658  006616  052777  001000  172552          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
2659  006624  017737  172550  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
2660                                                                  ;/PREVIOUS COUNTER
2661  006632  012677  172540                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
2662  006636  005737  001126                  TST     $BDDAT          ;/RESTORE CSR
2663  006642  001003                          BNE     2$              ;IF ANY COUNT, ERROR.
2664  006644  105777  172526                  TSTB    @ASR            ;CHECK FOR OVERFLOW.
2665  006650  100003                          BPL     3$              ;IF NO OVERFLOW, NEXT TEST
2666
2667  006652  005037  001124          2$:     CLR     $GDDAT          ;CLEAR FOR TYPEOUT, EXP'D ZERO.
2668

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

2671  006656  104011                  ERROR   11              ;CLOCK COUNTED WHEN RATE:"SLAVE IN"
2672                                                          ;WAS SELECTED.
2673

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

2676  006660  005077  172512          3$:     CLR     @ASR            ;CLEAR CLOCK'S CSR.
2677
2678
2679                                    ;;**************************************************************
2680                                    ;*TEST 43       *TEST THAT THE CLOCK WILL COUNT IN MODE 1
2681                                    ;;**************************************************************
2682  006664  000004                  TST43:  SCOPE
2683
2684  006666  005077  172504                  CLR     @ASR            ;CLEAR THE CSR.
2685  006672  012777  177777  172500          MOV     #-1,@ABR        ;PRESET BUFFER REG.
2686  006700  052777  000013  172470          BIS     #13,@ASR        ;START CLOCK,RATE:1 MHZ,MODE 1.
2687  006706  005000                          CLR     R0
2688  006710  105200                  1$:     INCB    R0              ;NOW DO A SHORT DELAY SO THAT THE
2689  006712  001376                          BNE     1$              ;CLOCK CAN COUNT TO OVERFLOW.
2690
2691  006714  105777  172456                  TSTB    @ASR            ;OVERFLOW SHOULD HAVE SET.
2692  006720  100401                          BMI     2$              ;GOTO NEXT TEST IF SO.
2693
2694

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

2697  006722  104006                  ERROR   6               ;CLOCK FAILED TO COUNT UP AND SET
2698                                                          ;OVERFLOW IN MODE 1.
2699

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

2702  006724                          2$:
2703
2704
2705                                    .SBTTL  *
```

```
                                        .SBTTL  *PHASE 4 CLOCK INTERRUPT TEST.
                                        .SBTTL  *


                                ;;**********************************************************************
                                ;*TEST 44        *TEST THAT THE CLOCK WILL INTERRUPT ON OVERFLOW
                                ;;**********************************************************************
006724  000004                  TST44:  SCOPE
006726  012737  000020  001160          MOV     #20,$TIMES      ;;DO 20 ITERATIONS

006734  012746  000340                  MOV     #340,-(SP)      ;PUT PRIORITY ON STACK.
006740  012746  006746                  MOV     #64$,-(SP)      ;PUT RETURN ADDRESS ON STACK
006744  000002                          RTI                     ;DO AN RTI, PUTS PRIORITY IN CPU.
006746                          64$:

006746  005077  172424                  CLR     @ASR            ;CLEAR CLOCK'S CSR.
006752  012777  177777  172420          MOV     #-1,@ABR        ;SET PRESET BUFFER TO ALL ONES.

006760  012777  000161  172410          MOV     #161,@ASR       ;START CLOCK, RATE:ST1.
006766  052777  000400  172402          BIS     #BIT8,@ASR      ;GENERATE A MAINTENANCE ST1.
006774  012777  007034  172400          MOV     #1$,@VECT1      ;SET INTERRUPT ADDR.

007002  012746  000000                  MOV     #0,-(SP)        ;PUT PRIORITY ON STACK.
007006  012746  007014                  MOV     #65$,-(SP)      ;PUT RETURN ADDRESS ON STACK
007012  000002                          RTI                     ;DO AN RTI, PUTS PRIORITY IN CPU.
007014                          65$:

007014  000240                          NOP                     ;STALL TIME

007016  012746  000340                  MOV     #340,-(SP)      ;PUT PRIORITY ON STACK.
007022  012746  007030                  MOV     #66$,-(SP)      ;PUT RETURN ADDRESS ON STACK
007026  000002                          RTI                     ;DO AN RTI, PUTS PRIORITY IN CPU.
007030                          66$:

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

007030  104007                          ERROR   7               ;CLOCK FAILED TO INTERRUPT.

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

007032  000402                          BR      2$
007034                          1$:
007034  062706  000004                  ADD     #4,SP                   ;/ADD #4 TO STACK POINTER.
007040  005077  172332          2$:     CLR     @ASR            ;CLEAR THE CLOCK.

                                ;;**********************************************************************
                                ;*TEST 45        *TEST THAT ST2 WILL CAUSE AN INTERRUPT
                                ;;**********************************************************************
007044  000004                  TST45:  SCOPE

007046  012746  000340                  MOV     #340,-(SP)      ;PUT PRIORITY ON STACK.
007052  012746  007060                  MOV     #64$,-(SP)      ;PUT RETURN ADDRESS ON STACK
```

```
007056  000002                                  RTI                             ;DO AN RTI, PUTS PRIORITY IN CPU.
007060                                  64$:

007060  005077  172312                          CLR     @ASR                    ;CLEAR CLOCKS CSR.
007064  012777  007140  172314                  MOV     #1$,@VECT2              ;SET UP INTERRUPT VECTOR.
007072  012777  040001  172276                  MOV     #BIT14!BIT0,@ASR;SET "INT2",AND GO BIT.
007100  052777  001000  172270                  BIS     #BIT9,@ASR              ;GENERATE A MAINTENANCE ST2.

007106  012746  000000                          MOV     #0,-(SP)                ;PUT PRIORITY ON STACK.
007112  012746  007120                          MOV     #65$,-(SP)              ;PUT RETURN ADDRESS ON STACK
007116  000002                                  RTI                             ;DO AN RTI, PUTS PRIORITY IN CPU.
007120                                  65$:

007120  000240                                  NOP                             ;STALL TIME

007122  012746  000340                          MOV     #340,-(SP)              ;PUT PRIORITY ON STACK.
007126  012746  007134                          MOV     #66$,-(SP)              ;PUT RETURN ADDRESS ON STACK
007132  000002                                  RTI                             ;DO AN RTI, PUTS PRIORITY IN CPU.
007134                                  66$:

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

007134  104007                                  ERROR   7                       ;CLOCK FAILED TO INTERRUPT ON ST2.

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

007136  000402                                  BR      2$
007140                                  1$:
007140  062706  000004                          ADD     #4,SP                           ;/ADD #4 TO STACK POINTER.
007144  005077  172226                  2$:     CLR     @ASR                    ;CLEAR CLOCK'S CSR.

                                                .SBTTL  *
                                                .SBTTL  *PHASE 5 ADVANCED TESTING
                                                .SBTTL  *
```

```
2795                                    ;;**********************************************************************
2796                                    ;*TEST 46       *TEST T4AT THE "FOR" BIT WILL SET ON 2 ST2'S
2797                                    ;;**********************************************************************
2798  007150  000004                   T$T46:  SCOPE
2799
2800  007152  005077  172220                   CLR     @ASR            ;START WITH CSR CLEAR.
2801  007156  005277  172214                   INC     @ASR            ;SET GO BIT.
2802  007162  052777  001000  172206           BIS #BIT9,@ASR          ; GENERATE THE 1ST ST2 PULSE.
2803  007170  052777  001000  172200           BIS     #BIT9,@ASR      ;GENERATE 2ND ST2 PULSE.
2804                                                                    ;THIS SHOULD CAUSE FOR BIT TO SET.
2805  007176  032777  010000  172172           BIT     #BIT12,@ASR     ;DID FOR BIT SET?
2806  007204  001007                           BNE     1$              ;YES-THEN NEXT TEST.
2807
2808  007206  017737  172164  001126           MOV     @ASR,$BDDAT     ;RECORD CSR.
2809  007214  012737  110001  001124           MOV     #BIT15!BIT12!BIT0,$GDDAT        ;RECORD S/B.
2810
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2813  007222  104001                           ERROR   1               ;ERROR-"FOR" BIT FAILED TO SET ON
2814                                                                    ;ON TWO SUCCESIVE ST2 PULSES.
2815
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2818  007224                           1$:
2819
2820                                    ;;**********************************************************************
2821                                    ;*TEST 47       *TEST THAT FOR BIT WILL SET ON TWO OVERFLOWS
2822                                    ;;**********************************************************************
2823  007224  000004                   T$T47:  SCOPE
2824  007226  012737  000002  001160           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
2825
2826  007234  005077  172136                   CLR     @ASR            ;START WITH CSR CLEAR.
2827  007240  012777  177777  172132           MOV     #-1,@ABR        ;PRELOAD BUFFER REG.
2828  007246  052777  000013  172122           BIS     #13,@ASR        ;START CLOCK,MODE 1,1MHZ.
2829  007254  005000                           CLR     R0              ;NOW DO A SHORT DELAY.
2830  007256  105200                   1$:     INCB    R0              ;DURING THIS DELAY, THE CLOCK WILL
2831  007260  001376                           BNE     1$              ;HAVE OVERFLOWED TWICE-
2832                                                                    ;THIS SHOULD CAUSE THE FOR BIT TO SET.
2833
2834  007262  032777  010000  172106           BIT     #BIT12,@ASR     ;DID FOR SET?
2835  007270  001007                           BNE     2$              ;YES-NEXT TEST.
2836
2837  007272  017737  172100  001126           MOV     @ASR,$BDDAT     ;NO-RECORD THE CSR.
2838  007300  012737  010213  001124           MOV     #010213,$GDDAT  ;RECORD S/B.
2839
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2842  007306  104001                           ERROR   1               ;ERROR FOR BIT FAILED TO SET ON
2843                                                                    ; TWO SUCCESSIVE OVERFLOWS.
2844
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2847  007310                           2$:
2848                                    ;;**********************************************************************
```

```
2849                                    ;*TEST 50       *TEST THAT FOR BIT WILL CLEAR IF GO BIT IS SET
2850                                    ;;**************************************************************
2851  007310  000004                   T$T50:  SCOPE
2852
2853  007312  012777  000001  172056          MOV     #BIT0,@ASR      ;CLEAR CSR,SET GO BIT.
2854  007320  052777  001000  172050          BIS     #BIT9,@ASR      ;SET 1ST ST2 PULSE.
2855  007326  052777  001000  172042          BIS     #BIT9,@ASR      ;GENERATE 2ND ST2 PULSE.
2856                                                                  ;FOR BIT SETS HERE.
2857  007334  042777  000001  172034          BIC     #BIT0,@ASR      ;CLEAR GO BIT.
2858
2859  007342  052777  000001  172026          BIS     #BIT0,@ASR      ;SET THE "GO" BIT AGAIN -
2860                                                                  ;SHOULD CLEAR FOR BIT.
2861
2862  007350  017737  172022  001126          MOV     @ASR,$BDDAT     ;READ THE CSR.
2863
2864  007356  012737  100001  001124          MOV     #100001,$GDDAT  ;RECORD WHAT CSR S/B.
2865  007364  032737  010000  001126          BIT     #BIT12,$BDDAT   ;DID FOR BIT CLEAR?
2866  007372  001401                          BEQ     1$              ;YES NEXT TEST.
2867
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2870  007374  104001                          ERROR   1               ;FOR BIT FALED TO CLEAR
2871                                                                  ;WHEN "GO" BIT WAS SET.
2872
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

2875  007376  005077  171774          1$:     CLR     @ASR            ;CLEAR THE CSR.
2876
2877                                    ;;**************************************************************
2878                                    ;*TEST 51       *TEST THAT WE CAN DISABLE THE INTERNAL OSC
2879                                    ;;**************************************************************
2880  007402  000004                   T$T51:  SCOPE
2881  007404  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2882
2883  007412  005077  171760                  CLR     @ASR            ;CLEAR THE CSR
2884  007416  005077  171756                  CLR     @ABR            ;CLEAR THE PRESET BUFFER
2885  007422  005037  001124                  CLR     $GDDAT          ;CLEAR EXPED.
2886
2887  007426  012777  004000  171742          MOV     #BIT11,@ASR     ;DISABLE THE INTERNAL OSC.
2888  007434  052777  000011  171734          BIS     #BIT3!BIT0,@ASR ;START CLOCK:RATE 1MHZ.
2889  007442  005000                          CLR     R0
2890  007444  105200                  1$:     INCB    R0              ;DELAY A SHORT TIME.
2891  007446  001376                          BNE     1$
2892  007450  017746  171722                  MOV     @ASR,-(6)       ;/SAVE CSR
2893  007454  011637  001424                  MOV     (6),$TMP3       ;/GET CSR.
2894  007460  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
2895  007466  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2896  007474  013777  001424  171674          MOV     $TMP3,@ASR              ;/LOAD CSR.
2897                                                                  ;/THIS MUST BE DONE IN
2898                                                                  ;/ORDER TO XFERR COUNTER
2899                                                                  ;/TO BUFFER ON ST2.
2900  007502  052777  001000  171666          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
2901  007510  017737  171664  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
2902                                                                  ;/PREVIOUS COUNTER
```

```
007516  012677  171654                   MOV     (6)+,@ASR          ;/CONTENTS ARE IN $BDDAT.
007522  005737  001126                   TST     $BDDAT             ;/RESTORE CSR
007526  001401                           BEQ     2$                 ;NO - GOOD - NEXT TEST.

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

007530  104011                           ERROR   11                 ;CLOCK DISABLE INTERNAL
                                                                    ;OSC. DID NOT WORK.

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

007532  005077  171640           2$:     CLR     @ASR               ;CLEAR THE CSR.

                                 ;;*************************************************************
                                 ;*TEST 52       *TEST THAT CLOCK CAN BE COUNTED USING MAINTENANCE OSC
                                 ;;*************************************************************
007536  000004                   TST52:  SCOPE

007540  005077  171632                   CLR     @ASR               ;CLEAR THE CSR.
007544  005077  171630                   CLR     @ABR               ;CLEAR THE PRESET BUFFER.
007550  052777  004000  171620           BIS     #BIT11,@ASR        ;DISABLE THE INTERNAL OSC.
007556  052777  000011  171612           BIS     #BIT3!BIT0,@ASR    ;START CLOCK, 1MHZ, GO.
007564  012700  177754                   MOV     #-20.,R0           ;SET TO COUNT 20 TIMES

007570  052777  002000  171600   1$:     BIS     #BIT10,@ASR        ;GENERATE 1 MAINTENANCE OSC.
                                                                    ;NOTE: AT 1MHZ, IT TAKES 10
                                                                    ;MAINT. OSC TO EQUAL 1 COUNT
007576  005200                           INC     R0                 ;DO 20 MAINTENANCE OSC.
007600  001373                           BNE     1$

007602  017746  171570                   MOV     @ASR,-(6)          ;/SAVE CSR
007606  011637  001424                   MOV     (6),$TMP3          ;/GET CSR.
007612  042737  177707  001424           BIC     #177707,$TMP3      ;/SAVE RATE BITS.
007620  052737  004005  001424           BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
007626  013777  001424  171542           MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                    ;/THIS MUST BE DONE IN
                                                                    ;/ORDER TO XFERR COUNTER
                                                                    ;/TO BUFFER ON ST2.
007634  052777  001000  171534           BIS     #BIT9,@ASR         ;/GENERATE ON ST2 PULSE
007642  017737  171532  001126           MOV     @ABR,$BDDAT        ;/READ THE PRESET BUFFER,
                                                                    ;/PREVIOUS COUNTER
007650  012677  171522                   MOV     (6)+,@ASR          ;/CONTENTS ARE IN $BDDAT.
007654  005737  001126                   TST     $BDDAT             ;/RESTORE CSR
007660  001001                           BNE     2$                 ;YES - NEXT TEST.

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

007662  104011                           ERROR   11                 ;ERROR COULD NOT COUNT USING
                                                                    ;MAINTENANCE OSC.

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

007664  005077  171506           2$:     CLR     @ASR               ;CLEAR THE CSR.
```

```
                                        ;;**********************************************************************
                                        ;*TEST 53        *TEST THE CLOCK'S 1MHZ DIVIDER
                                        ;;**********************************************************************
007670  000004                  TST53:  SCOPE
007672  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

007700  005077  171472                  CLR     @ASR            ;/CLEAR THE CSR.
007704  005077  171470                  CLR     @ABR            ;/CLEAR THE PRESET BUFFER.
007710  052777  004000  171460          BIS     #BIT11,@ASR     ;/DISABLE THE INTERNAL OSC.
007716  052777  000011  171452          BIS     #1!10,@ASR      ;/ENABLE CLOCK, RATE:1MHZ

007724  012700  177766          10$:    MOV     #-10.,R0        ;/SET TO GENERATE 10 OSC PULSES.

007730  052777  002000  171440  1$:     BIS     #BIT10,@ASR     ;/GENERATE ONE OSC PULSE.
007736  005200                          INC     R0              ;/DONE 10 OSC PULSES?
007740  001373                          BNE     1$              ;/NO - DO ANOTHER ONE.

007742  012737  000001  001124  2$:     MOV     #1,$GDDAT       ;/SET FOR ERROR TYPEOUT - IF ANY.
007750  017746  171422                  MOV     @ASR,-(6)       ;/SAVE CSR
007754  011637  001424                  MOV     (6),$TMP3       ;/GET CSR.
007760  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
007766  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
007774  013777  001424  171374          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                ;/THIS MUST BE DONE IN
                                                                ;/ORDER TO XFERR COUNTER
                                                                ;/TO BUFFER ON ST2.
010002  052777  001000  171366          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
010010  017737  171364  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
                                                                ;/PREVIOUS COUNTER
010016  012677  171354                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
010022  005737  001126                  TST     $BDDAT          ;/RESTORE CSR
010026  013737  001124  001420          MOV     $GDDAT,$TMP0    ;/$TMP0 USED IN ERROR TYPEOUT.

010034  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID CLOCK ADVANCE ONCE?

010042  001402                          BEQ     3$              ;/YES - NEXT TEST.

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

010044  104011                          ERROR   11              ;/ERROR ON CLOCK1MHZ PULSE
                                                                ;/NOT GENERATED WHEN 10
                                                                ;/OSC PULSES GENERATED.

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

010046  000443                          BR      5$
010050  012700  000011          3$:     MOV     #9.,R0  ;/GET THE NUMBER OF MORE OSC PULSES
                                                        ;/TO BE GENERATED.
```

```
010054  052777  002000  171314  4$:     BIS     #BIT10,@ASR             ;/GENERATE ANOTHER OSC PULSE.
010062  005300                          DEC     R0                      ;/WHAT WE WANT TO CHECK
010064  001373                          BNE     4$                      ;/1MHZ PULSE ON 9 OSC PULSES.


010066  017746  171304                  MOV     @ASR,-(6)               ;/SAVE CSR
010072  011637  001424                  MOV     (6),$TMP3               ;/GET CSR.
010076  042737  177707  001424          BIC     #177707,$TMP3           ;/SAVE RATE BITS.
010104  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
010112  013777  001424  171256          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                        ;/THIS MUST BE DONE IN
                                                                        ;/ORDER TO XFERR COUNTER
                                                                        ;/TO BUFFER ON ST2.
010120  052777  001000  171250          BIS     #BIT9,@ASR              ;/GENERATE ON ST2 PULSE
010126  017737  171246  001126          MOV     @ABR,$BDDAT             ;/READ THE PRESET BUFFER,
                                                                        ;/PREVIOUS COUNTER
010134  012677  171236                  MOV     (6)+,@ASR               ;/CONTENTS ARE IN $BDDAT.
010140  005737  001126                  TST     $BDDAT                  ;/RESTORE CSR
010144  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;/WAS ANOTHER 1MHZ PULSE GENERATED?
010152  001401                          BEQ     5$                      ;/NO - NEXT TEST.

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

010154  104011                          ERROR   11                      ;/WE SEEM TO HAVE GENERATED
                                                                        ;/ANOTHER 1MHZ PULSE ON
                                                                        ;/ONLY 9 MAINTENANCE
                                                                        ;/OSC PULSES.

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

010156  005077  171214          5$:     CLR     @ASR                    ;/CLEAR THE CSR.



                                ;;*************************************************************************
                                ;*TEST 54        *TEST THE CLOCK'S 100KHZ DIVIDER
                                ;;*************************************************************************
010162  000004                  TST54:  SCOPE
010164  012737  000005  001160          MOV     #5,$TIMES               ;;DO 5 ITERATIONS

010172  005077  171200                  CLR     @ASR                    ;/CLEAR THE CSR.
010176  005077  171176                  CLR     @ABR                    ;/CLEAR THE PRESET BUFFER.
010202  052777  004000  171166          BIS     #BIT11,@ASR             ;/DISABLE THE INTERNAL OSC.
010210  052777  000021  171160          BIS     #1!20,@ASR              ;/ENABLE CLOCK, RATE:100KHZ


010216  012700  177634          10$:    MOV     #-100.,R0               ;/SET TO GENERATE 100 OSC PULSES.

010222  052777  002000  171146  1$:     BIS     #BIT10,@ASR             ;/GENERATE ONE OSC PULSE.
010230  005200                          INC     R0                      ;/DONE 100 OSC PULSES?
010232  001373                          BNE     1$                      ;/NO - DO ANOTHER ONE.


010234  012737  000001  001124  2$:     MOV     #1,$GDDAT               ;/SET FOR ERROR TYPEOUT - IF ANY.
010242  017746  171130                  MOV     @ASR,-(6)               ;/SAVE CSR
```

```
3065  010246  011637  001424                MOV   (6),$TMP3       ;/GET CSR.
3066  010252  042737  177707  001424        BIC   #177707,$TMP3   ;/SAVE RATE BITS.
3067  010260  052737  004005  001424        BIS   #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3068  010266  013777  001424  171102        MOV   $TMP3,@ASR              ;/LOAD CSR.
3069                                                                      ;/THIS MUST BE DONE IN
3070                                                                      ;/ORDER TO XFERR COUNTER
3071                                                                      ;/TO BUFFER ON ST2.
3072  010274  052777  001000  171074        BIS   #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
3073  010302  017737  171072  001126        MOV   @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
3074                                                              ;/PREVIOUS COUNTER
3075  010310  012677  171062                MOV   (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
3076  010314  005737  00112E                TST   $BDDAT          ;/RESTORE CSR
3077  010320  013737  001124  001420        MOV   $GDDAT,$TMP0    ;/$TMP0 USED IN ERROR TYPEOUT.
3078
3079  010326  023737  001124  001126        CMP   $GDDAT,$BDDAT   ;/DID CLOCK ADVANCE ONCE?
3080
3081  010334  001402                        BEQ   3$              ;/YES - NEXT TEST.
3082

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

3085  010336  104011                        ERROR 11              ;/ERROR ON CLOCK100KHZ PULSE
3086                                                              ;/NOT GENERATED WHEN 100
3087                                                              ;/OSC PULSES GENERATED.
3088

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

3091  010340  000443                        BR    5$
3092  010342  012700  000143          3$:   MOV   #99.,R0 ;/GET THE NUMBER OF MORE OSC PULSES
3093                                                      ;/TO BE GENERATED.
3094
3095  010346  052777  002000  171022  4$:   BIS   #BIT10,@ASR      ;/GENERATE ANOTHER OSC PULSE.
3096  010354  005300                        DEC   R0              ;/WHAT WE WANT TO CHECK
3097  010356  001373                        BNE   4$              ;/100KHZ PULSE ON 99 OSC PULSES.
3098
3099
3100  010360  017746  171012                MOV   @ASR,-(6)       ;/SAVE CSR
3101  010364  011637  001424                MOV   (6),$TMP3       ;/GET CSR.
3102  010370  042737  177707  001424        BIC   #177707,$TMP3   ;/SAVE RATE BITS.
3103  010376  052737  004005  001424        BIS   #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3104  010404  013777  001424  170764        MOV   $TMP3,@ASR              ;/LOAD CSR.
3105                                                                      ;/THIS MUST BE DONE IN
3106                                                                      ;/ORDER TO XFERR COUNTER
3107                                                                      ;/TO BUFFER ON ST2.
3108  010412  052777  001000  170756        BIS   #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
3109  010420  017737  170754  001126        MOV   @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
3110                                                              ;/PREVIOUS COUNTER
3111  010426  012677  170744                MOV   (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
3112  010432  005737  001126                TST   $BDDAT          ;/RESTORE CSR
3113  010436  023737  001124  001126        CMP   $GDDAT,$BDDAT   ;/WAS ANOTHER 100KHZ PULSE GENERATED?
3114  010444  001401                        BEQ   5$              ;/NO - NEXT TEST.
3115

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

3118  010446  104011                        ERROR 11              ;/WE SEEM TO HAVE GENERATED
```

```
                                                                        ;/ANOTHER 100KHZ PULSE ON
                                                                        ;/ONLY 99 MAINTENANCE
                                                                        ;/OSC PULSES.

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

010450  005077  170722          5$:     CLR     @ASR                    ;/CLEAR THE CSR.


                                        ;;*****************************************************************
                                        ;*TEST 55       *TEST THE CLOCK'S 10KHZ DIVIDER
                                        ;;*****************************************************************
010454  000004                  TST55:  SCOPE
010456  012737  000005  001160          MOV     #5,$TIMES               ;;DO 5 ITERATIONS

010464  005077  170706                  CLR     @ASR                    ;/CLEAR THE CSR.
010470  005077  170704                  CLR     @ABR                    ;/CLEAR THE PRESET BUFFER.
010474  052777  004000  170674          BIS     #BIT11,@ASR             ;/DISABLE THE INTERNAL OSC.
010502  052777  000031  170666          BIS     #1!30,@ASR              ;/ENABLE CLOCK, RATE:10KHZ


010510  012700  176030          10$:    MOV     #-1000.,R0              ;/SET TO GENERATE 1000 OSC PULSES.

010514  052777  002000  170654  1$:     BIS     #BIT10,@ASR             ;/GENERATE ONE OSC PULSE.
010522  005200                          INC     R0                      ;/DONE 1000 OSC PULSES?
010524  001373                          BNE     1$                      ;/NO - DO ANOTHER ONE.


010526  012737  000001  001124  2$:     MOV     #1,$GDDAT               ;/SET FOR ERROR TYPEOUT - IF ANY.
010534  017746  170636                  MOV     @ASR,-(6)               ;/SAVE CSR
010540  011637  001424                  MOV     (6),$TMP3               ;/GET CSR.
010544  042737  177707  001424          BIC     #177707,$TMP3           ;/SAVE RATE BITS.
010552  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3    ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
010560  013777  001424  170610          MOV     $TMP3,@ASR                ;/LOAD CSR.
                                                                        ;/THIS MUST BE DONE IN
                                                                        ;/ORDER TO XFERR COUNTER
                                                                        ;/TO BUFFER ON ST2.
010566  052777  001000  170602          BIS     #BIT9,@ASR              ;/GENERATE ON ST2 PULSE
010574  017737  170600  001126          MOV     @ABR,$BDDAT             ;/READ THE PRESET BUFFER,
                                                                        ;/PREVIOUS COUNTER
010602  012677  170570                  MOV     (6)+,@ASR               ;/CONTENTS ARE IN $BDDAT.
010606  005737  001126                  TST     $BDDAT                  ;/RESTORE CSR
010612  013737  001124  001420          MOV     $GDDAT,$TMP0            ;/$TMP0 USED IN ERROR TYPEOUT.

010620  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;/DID CLOCK ADVANCE ONCE?

010626  001402                          BEQ     3$                      ;/YES - NEXT TEST.

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

010630  104011                          ERROR   11                      ;/ERROR ON CLOCK10KHZ PULSE
                                                                        ;/NOT GENERATED WHEN 1000
                                                                        ;/OSC PULSES GENERATED.

```

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

010632  000443                                  BR      5$
010634  012700  001747          3$:     MOV     #999.,R0        ;/GET THE NUMBER OF MORE OSC PULSES
                                                                ;/TO BE GENERATED.

010640  052777  002000  170530  4$:     BIS     #BIT10,@ASR     ;/GENERATE ANOTHER OSC PULSE.
010646  005300                          DEC     R0              ;/WHAT WE WANT TO CHECK
010650  001373                          BNE     4$              ;/10KHZ PULSE ON 999 OSC PULSES.


010652  017746  170520                  MOV     @ASR,-(6)       ;/SAVE CSR
010656  011637  001424                  MOV     (6),$TMP3       ;/GET CSR.
010662  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
010670  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
010676  013777  001424  170472          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                ;/THIS MUST BE DONE IN
                                                                ;/ORDER TO XFERR COUNTER
                                                                ;/TO BUFFER ON ST2.
010704  052777  001000  170464          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
010712  017737  170462  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
                                                                ;/PREVIOUS COUNTER
010720  012677  170452                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
010724  005737  001126                  TST     $BDDAT          ;/RESTORE CSR
010730  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/WAS ANOTHER 10KHZ PULSE GENERATED?
010736  001401                          BEQ     5$              ;/NO - NEXT TEST.

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

010740  104011                          ERROR   11              ;/WE SEEM TO HAVE GENERATED
                                                                ;/ANOTHER 10KHZ PULSE ON
                                                                ;/ONLY 999 MAINTENANCE
                                                                ;/OSC PULSES.

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

010742  005077  170430          5$:     CLR     @ASR            ;/CLEAR THE CSR.




                                        ;;*************************************************************
                                        ;*TEST 56        *TEST THE CLOCK'S 1KHZ DIVIDER
                                        ;;*************************************************************
010746  000004                  TST56:  SCOPE
010750  012737  000005  001160          MOV     #5,$TIMES       ;;DO 5 ITERATIONS

010756  005077  170414                  CLR     @ASR            ;/CLEAR THE CSR.
010762  005077  170412                  CLR     @ABR            ;/CLEAR THE PRESET BUFFER.
010766  052777  004000  170402          BIS     #BIT11,@ASR     ;/DISABLE THE INTERNAL OSC.
010774  052777  000041  170374          BIS     #1!40,@ASR      ;/ENABLE CLOCK, RATE:1KHZ


011002  012700  154360          10$:    MOV     #-10000.,R0     ;/SET TO GENERATE 10000 OSC PULSES.

```

```
011006  052777  002000  170362  1$:     BIS     #BIT10,@ASR             ;/GENERATE ONE OSC PULSE.
011014  005200                          INC     R0                      ;/DONE 10000 OSC PULSES?
011016  001373                          BNE     1$                      ;/NO - DO ANOTHER ONE.


011020  012737  000001  001124  2$:     MOV     #1,$GDDAT               ;/SET FOR ERROR TYPEOUT - IF ANY.
011026  017746  170344                  MOV     @ASR,-(6)               ;/SAVE CSR
011032  011637  001424                  MOV     (6),$TMP3               ;/GET CSR.
011036  042737  177707  001424          BIC     #177707,$TMP3           ;/SAVE RATE BITS.
011044  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
011052  013777  001424  170316          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                        ;/THIS MUST BE DONE IN
                                                                        ;/ORDER TO XFERR COUNTER
                                                                        ;/TO BUFFER ON ST2.
011060  052777  001000  170310          BIS     #BIT9,@ASR              ;/GENERATE ON ST2 PULSE
011066  017737  170306  001126          MOV     @ABR,$BDDAT             ;/READ THE PRESET BUFFER,
                                                                        ;/PREVIOUS COUNTER
011074  012677  170276                  MOV     (6)+,@ASR               ;/CONTENTS ARE IN $BDDAT.
011100  005737  001126                  TST     $BDDAT                  ;/RESTORE CSR
011104  013737  001124  001420          MOV     $GDDAT,$TMP0            ;/$TMP0 USED IN ERROR TYPEOUT.

011112  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;/DID CLOCK ADVANCE ONCE?

011120  001402                          BEQ     3$                      ;/YES - NEXT TEST.

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

011122  104011                          ERROR   11                      ;/ERROR ON CLOCK1KHZ PULSE
                                                                        ;/NOT GENERATED WHEN 10000
                                                                        ;/OSC PULSES GENERATED.

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

011124  000443                          BR      5$
011126  012700  023417          3$:     MOV     #9999.,R0               ;/GET THE NUMBER OF MORE OSC PULSES
                                                                        ;/TO BE GENERATED.

011132  052777  002000  170236  4$:     BIS     #BIT10,@ASR             ;/GENERATE ANOTHER OSC PULSE.
011140  005300                          DEC     R0                      ;/WHAT WE WANT TO CHECK
011142  001373                          BNE     4$                      ;/1KHZ PULSE ON 9999 OSC PULSES.


011144  017746  170226                  MOV     @ASR,-(6)               ;/SAVE CSR
011150  011637  001424                  MOV     (6),$TMP3               ;/GET CSR.
011154  042737  177707  001424          BIC     #177707,$TMP3           ;/SAVE RATE BITS.
011162  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
011170  013777  001424  170200          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                        ;/THIS MUST BE DONE IN
                                                                        ;/ORDER TO XFERR COUNTER
                                                                        ;/TO BUFFER ON ST2.
011176  052777  001000  170172          BIS     #BIT9,@ASR              ;/GENERATE ON ST2 PULSE
011204  017737  170170  001126          MOV     @ABR,$BDDAT             ;/READ THE PRESET BUFFER,
                                                                        ;/PREVIOUS COUNTER
011212  012677  170160                  MOV     (6)+,@ASR               ;/CONTENTS ARE IN $BDDAT.
```

```
011216  005737  001126                    TST     $BDDAT            ;/RESTORE CSR
011222  023737  001124  001126            CMP     $GDDAT,$BDDAT     ;/WAS ANOTHER 1KHZ PULSE GENERATED?
011230  001401                            BEQ     5$                ;/NO - NEXT TEST.

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

011232  104011                            ERROR   11                ;/WE SEEM TO HAVE GENERATED
                                                                    ;/ANOTHER 1KHZ PULSE ON
                                                                    ;/ONLY 9999 MAINTENANCE
                                                                    ;/OSC PULSES.

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

011234  005077  170136            5$:     CLR     @ASR              ;/CLEAR THE CSR.


                                  ;;*************************************************************
                                  ;*TEST 57        *TEST THE CLOCK'S 100HZ DIVIDER
                                  ;;*************************************************************
011240  000004                    TST57:  SCOPE
011242  012737  000005  001160            MOV     #5,$TIMES         ;;DO 5 ITERATIONS

011250  005077  170122                    CLR     @ASR              ;/CLEAR THE CSR.
011254  005077  170120                    CLR     @ABR              ;/CLEAR THE PRESET BUFFER.
011260  052777  004000  170110            BIS     #BIT11,@ASR       ;/DISABLE THE INTERNAL OSC.
011266  052777  000051  170102            BIS     #1!50,@ASR        ;/ENABLE CLOCK, RATE:100HZ

011274  012701  000012                    MOV     #10.,R1

011300  012700  154360            10$:    MOV     #-10000.,R0       ;/SET TO GENERATE 10000 OSC PULSES.

011304  052777  002000  170064    1$:     BIS     #BIT10,@ASR       ;/GENERATE ONE OSC PULSE.
011312  005200                            INC     R0                ;/DONE 10000 OSC PULSES?
011314  001373                            BNE     1$                ;/NO - DO ANOTHER ONE.

011316  005301                            DEC     R1
011320  001367                            BNE     10$

011322  012737  000001  001124    2$:     MOV     #1,$GDDAT         ;/SET FOR ERROR TYPEOUT - IF ANY.
011330  017746  170042                    MOV     @ASR,-(6)         ;/SAVE CSR
011334  011637  001424                    MOV     (6),$TMP3         ;/GET CSR.
011340  042737  177707  001424            BIC     #177707,$TMP3     ;/SAVE RATE BITS.
011346  052737  004005  001424            BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
011354  013777  001424  170014            MOV     $TMP3,@ASR        ;/LOAD CSR.
                                                                    ;/THIS MUST BE DONE IN
                                                                    ;/ORDER TO XFERR COUNTER
                                                                    ;/TO BUFFER ON ST2.
011362  052777  001000  170006            BIS     #BIT9,@ASR        ;/GENERATE ON ST2 PULSE
011370  017737  170004  001126            MOV     @ABR,$BDDAT       ;/READ THE PRESET BUFFER,
                                                                    ;/PREVIOUS COUNTER
011376  012677  167774                    MOV     (6)+,@ASR         ;/CONTENTS ARE IN $BDDAT.
011402  005737  001126                    TST     $BDDAT            ;/RESTORE CSR
011406  013737  001124  001420            MOV     $GDDAT,$TMP0      ;/$TMP0 USED IN ERROR TYPEOUT.
```

```
3335  011414  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DID CLOCK ADVANCE ONCE?
3336
3337  011422  001402                          BEQ     66$             ;/YES - NEXT TEST
3338
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3341  011424  104011                          ERROR   11              ;/ERROR ON CLOCK100HZ PULSE
3342                                                                  ;/NOT GENERATED WHEN 10000
3343                                                                  ;/OSC PULSES GENERATED.
3344
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3347  011426  000447                          BR      5$
3348  011430  012701  000012          66$:    MOV     #10.,R1
3349  011434  012700  023417          3$:     MOV     #9999.,R0       ;/GET THE NUMBER OF MORE OSC PULSES
3350                                                                  ;/TO BE GENERATED.
3351
3352  011440  052777  002000  167730  4$:     BIS     #BIT10,@ASR     ;/GENERATE ANOTHER OSC PULSE.
3353  011446  005300                          DEC     R0              ;/WHAT WE WANT TO CHECK
3354  011450  001373                          BNE     4$              ;/100HZ PULSE ON 9999 OSC PULSES.
3355  011452  005301                          DEC     R1
3356  011454  001367                          BNE     3$
3357
3358
3359  011456  017746  167714                  MOV     @ASR,-(6)       ;/SAVE CSR
3360  011462  011637  001424                  MOV     (6),$TMP3       ;/GET CSR.
3361  011466  042737  177707  001424          BIC     #177707,$TMP3   ;/SAVE RATE BITS.
3362  011474  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3363  011502  013777  001424  167666          MOV     $TMP3,@ASR              ;/LOAD CSR.
3364                                                                  ;/THIS MUST BE DONE IN
3365                                                                  ;/ORDER TO XFERR COUNTER
3366                                                                  ;/TO BUFFER ON ST2.
3367  011510  052777  001000  167660          BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
3368  011516  017737  167656  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
3369                                                                  ;/PREVIOUS COUNTER
3370  011524  012677  167646                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
3371  011530  005737  001126                  TST     $BDDAT          ;/RESTORE CSR
3372  011534  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/WAS ANOTHER 100HZ PULSE GENERATED?
3373  011542  001401                          BEQ     5$              ;/NO - NEXT TEST.
3374
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3377  011544  104011                          ERROR   11              ;/WE SEEM TO HAVE GENERATED
3378                                                                  ;/ANOTHER 100HZ PULSE ON
3379                                                                  ;/ONLY 9999 MAINTENANCE
3380                                                                  ;/OSC PULSES.
3381
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

3384  011546  005077  167624          5$:     CLR     @ASR            ;/CLEAR THE CSR.
3385
3386
3387
3388                                    ;;**************************************************************************
```

```
3389                                    ;*TEST 60       *TEST THE CLOCK'S MODE 2 OPERATION
3390                                            ;*
3391                                            ;*IN THIS TEST WE'LL CHECK MODE 2 OPERATION
3392                                            ;*MODE 2: EXTERNAL EVENTS TIMING MODE
3393                                            ;*SETTING THE GO BIT CAUSES THE COUNTER TO BEGIN COUNTING FROM
3394                                            ;*ZERO AND TO FREE-RUN UNTIL THE GO BIT IS WRITTEN
3395                                            ;*TO A ZERO THE COUNTER WILL CONTINUE COUNTING AFTER
3396                                            ;*OVERFLOW. AN EXTERNAL PULSE FROM SCHMITZ TRIGGER 2
3397                                            ;*(WHEN ST2 GO ENABLE IS A "0") CAUSES DATA TO
3398                                            ;*TRANSFER FROM THE COUNTER TO THE BUFFER/PRESET REG.
3399                                            ;*WHILE THE COUNTER CONTINUES TO RUN.
3400                                            ;*
3401                                            ;*TO TEST THIS MODE, WE'LL DISABLE THE INTERNAL OSC AND USE
3402                                            ;*MAINTENANCE OSC PULSES AS WELL AS A MAINTENANCE
3403                                            ;*ST2.
3404                                            ;*
3405                                    ;;**************************************************************
3406  011552  000004                    $TST60: SCOPE
3407  011554  012737  000020  001160            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
3408
3409  011562  005077  167610                    CLR     @ASR            ;CLEAR THE CSR.
3410  011566  005077  167606                    CLR     @ABR            ;CLEAR THE PRESET REG.
3411  011572  012777  004015  167576            MOV     #004015,@ASR    ;START CLOCK.
3412
3413  011600  012700  177754            1$:     MOV     #-20.,R0        ;SET TO GIVE 20 MAINTENANCE OSC.
3414  011604  052777  002000  167564    2$:     BIS     #BIT10,@ASR     ;GENERATE A MAINTENANCE OSC.
3415  011612  005200                            INC     R0
3416  011614  001373                            BNE     2$              ;IF NOT DONE 20 TIMES, LOOP.
3417
3418  011616  052777  001000  167552    3$:     BIS     #BIT9,@ASR      ;HERE'S THE BIGGIE! AN ST2 HAS BEEN GENERATED
3419  011624  012737  000002  001124            MOV     #2,$GODAT       ;THE PRESET BUFFER SHOULD BE 2.
3420  011632  017737  167542  001126            MOV     @ABR,$BDDAT     ;READ THE PRESET BUFFER.
3421  011640  023737  001126  001124            CMP     $BDDAT,$GODAT   ;DID A COUNTER TO PRESET BUFFER OCCUR?
3422  011646  001402                            BEQ     4$              ;YES - NEXT SUBTEST.
3423
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
3426  011650  104005                            ERROR   5               ;A COUNTER TO PRESET BUFFER DID NOT
3427                                                                    ;HAPPEN PROPERLY.
3428
                                        ;;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<<$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
3431  011652  000434                            BR      5$
3432
3433  011654                            4$:
3434  011654  017746  167516                    MOV     @ASR,-(6)       ;/SAVE CSR
3435  011660  011637  001424                    MOV     (6),$TMP3       ;/GET CSR.
3436  011664  042737  177707  001424            BIC     #177707,$TMP3   ;/SAVE RATE BITS.
3437  011672  052737  004005  001424            BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE.DISABLE INTERNAL OSC
3438  011700  013777  001424  167470            MOV     $TMP3,@ASR              ;/LOAD CSR.
3439                                                                    ;/THIS MUST BE DONE IN
3440                                                                    ;/ORDER TO XFERR COUNTER
3441                                                                    ;/TO BUFFER ON ST2.
3442  011706  052777  001000  167462            BIS     #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
```

```
3443  011714  017737  167460  001126          MOV     @ABR,$BDDAT     ;/READ THE PRESET BUFFER,
3444                                                                    ;/PREVIOUS COUNTER
3445  011722  012677  167450                  MOV     (6)+,@ASR       ;/CONTENTS ARE IN $BDDAT.
3446  011726  005737  001126                  TST     $BDDAT          ;/RESTORE CSR
3447  011732  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;WAS THE COUNTER ACCIDENTLY ZEROED?
3448  011740  001401                          BEQ     5$              ;NO - NEXT TEST.
3449

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

3452  011742  104005                          ERROR   5               ;THE COUNT REGISTER SHOULD NOT
3453                                                                    ;HAVE BEEN EFFECTED BY THE ST2
3454                                                                    ;IN MODE 2.
3455

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

3458  011744                            5$:


3459
3460
3461                                    ;;**************************************************************
3462                                    ;*TEST 61        *TEST THE CLOCK'S MODE 3 OPERATION
3463                                            ;*
3464                                            ;*IN THIS TEST WE'LL CHECK MODE 3 OPERATION.
3465                                            ;*MODE 3 IS JUST LIKE MODE 2 EXCEPT THAT THE COUNT
3466                                            ;*REG IS ZEROED AFTER AN ST2.
3467                                            ;*
3468                                    ;;**************************************************************
3469  011744  000004                    TST61:  SCOPE
3470  011746  012737  000020  001160          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
3471
3472  011754  005077  167416                  CLR     @ASR            ;CLEAR THE CSR.
3473  011760  005077  167414                  CLR     @ABR            ;CLEAR THE BUFFER REG.
3474  011764  012777  004017  167404          MOV     #4017,@ASR      ;START CLOCK.
3475
3476  011772  012700  177754          1$:     MOV     #-20.,R0        ;SET TO GIVE 20 MAINTENANCE OSC.
3477  011776  052777  002000  167372  2$:     BIS     #BIT10,@ASR     ;GENERATE A MAINTENANCE OSC.
3478  012004  005200                          INC     R0
3479  012006  001373                          BNE     2$              ;IF NOT DONE 20 TIMES, LOOP.
3480
3481  012010  052777  001000  167360  3$:     BIS     #BIT9,@ASR      ;HERE'S THE BIGGIE! AN ST2 HAS BEEN GENERATED
3482  012016  012737  000002  001124          MOV     #2,$GDDAT       ;THE PRESET BUFFER SHOULD BE 2.
3483  012024  017737  167350  001126          MOV     @ABR,$BDDAT     ;READ THE PRESET BUFFER.
3484  012032  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID A COUNTER TO PRESET BUFFER OCCUR?
3485  012040  001402                          BEQ     4$              ;YES - NEXT SUBTEST.
3486

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

3489  012042  104005                          ERROR   5               ;A COUNTER TO PRESET BUFFER DID NOT
3490                                                                    ;HAPPEN PROPERLY.
3491

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

3494  012044  000445                          BR      TST62           ;;
3495
3496  012046  005037  001124          4$:     CLR     $GDDAT          ;EXPECT ZERO BACK FROM COUNT REG.
```

```
012052  017746  167320                  MOV     @ASR,-(6)          ;/SAVE CSR
012056  011637  001424                  MOV     (6),$TMP3          ;/GET CSR.
012062  042737  177707  001424          BIC     #177707,$TMP3      ;/SAVE RATE BITS.
012070  052737  004005  001424          BIS     #BIT11!BIT2!BIT0,$TMP3  ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
012076  013777  001424  167272          MOV     $TMP3,@ASR              ;/LOAD CSR.
                                                                   ;/THIS MUST BE DONE IN
                                                                   ;/ORDER TO XFERR COUNTER
                                                                   ;/TO BUFFER ON ST2.
012104  052777  001000  167264          BIS     #BIT9,@ASR         ;/GENERATE ON ST2 PULSE
012112  017737  167262  001126          MOV     @ABR,$BDDAT        ;/READ THE PRESET BUFFER,
                                                                   ;/PREVIOUS COUNTER
012120  012677  167252                  MOV     (6)+,@ASR          ;/CONTENTS ARE IN $BDDAT.
012124  005737  001126                  TST     $BDDAT             ;/RESTORE CSR
012130  001402                          BEQ     5$                 ;IF SO - NEXT TEST.

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

012132  104005                          ERROR   5                  ;THE CLOCK FORGOT TO ZERO THE COUNT
                                                                   ;REG. AFTER AN ST2 OCCURRED ON
                                                                   ;A MODE 3 COUNT.

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

012134  000411                          BR      TST62              ;;
012136                          5$:
012136  005077  167234                  CLR     @ASR               ;NOW TRY CLEARING THE CSR.
012142  017737  167230  001126          MOV     @ASR,$BDDAT        ;READ THE CSR - DID IT CLEAR?
012150  001403                          BEQ     TST62              ;;
012152  005037  001124                  CLR     $GDDAT             ;NO - RECORD S/B.

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

012156  104002                          ERROR   2                  ;CSR FAILED TO CLEAR

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


                                ;;*******************************************************************
                                ;*TEST 62        *IF ENABLED,CHECK THRESHOLD ST1 FROM TESTOR
                                ;;*******************************************************************
012160  000004                  TST62:  SCOPE

012162  012737  000002  001160          MOV     #2,$TIMES          ;;DO 2 ITERATIONS.
012170  005737  001440                  TST     EXS                ;OPERATING IN TESTOR MODE?
012174  001002                          BNE     4$                 ;YES DO THIS TEST.
012176  000137  013016                  JMP     ENDP               ;NO-END PASS
012202                          4$:

012202  005077  167170                  CLR     @ASR               ;YES-CLEAR CSR.
012206  012777  177775  167164          MOV     #-3,@ABR           ;SET TO COUNT THREE TIMES.
012214  012777  000061  167154          MOV     #61,@ASR           ;SET RATE: ST1,MODE 1 GO.
012222  104401  012230                  TYPE    ,65$               ;;TYPE ASCIZ STRING
012226  000432                          BR      64$                ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <200><7><7>*SET ST1 THRESHOLD POT FULLY COUNTERCLOCKWIZE..*<7>
```

```
012314                                  64$:
012314  004737  013700                          JSR     PC,ANYKEY          ;TYPE LAST MESSAGE AND WAIT FOR OPERATER.
012320  012737  177775  001124                  MOV     #-3,$GDDAT         ;DON'T EXPECT COUNT TO CHANGE.
012326  017746  167044                          MOV     @ASR,-(6)          ;/SAVE CSR
012332  011637  001424                          MOV     (6),$TMP3          ;/GET CSR.
012336  042737  177707  001424                  BIC     #177707,$TMP3      ;/SAVE RATE BITS.
012344  052737  004005  001424                  BIS     #BIT11!BIT2!BIT0,$TMP3    ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
012352  013777  001424  167016                  MOV     $TMP3,@ASR                ;/LOAD CSR.
                                                                           ;/THIS MUST BE DONE IN
                                                                           ;/ORDER TO XFERR COUNTER
                                                                           ;/TO BUFFER ON ST2.
012360  052777  001000  167010                  BIS     #BIT9,@ASR         ;/GENERATE ON ST2 PULSE
012366  017737  167006  001126                  MOV     @ABR,$BDDAT        ;/READ THE PRESET BUFFER,
                                                                           ;/PREVIOUS COUNTER
012374  012677  166776                          MOV     (6)+,@ASR          ;/CONTENTS ARE IN $BDDAT.
012400  005737  001126                          TST     $BDDAT             ;/RESTORE CSR
012404  001002                                  BNE     2$      ;IF NON-ZERO,WE'RE OK.

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

012406  104002                                  ERROR   2                  ;COUNTERCLOCKWIZE THRESHOLD SETTING
                                                                           ;OF ST1 SHOULD HAVE INHIBITED
                                                                           ;ST1 FROM CAUSEING CLOCK TO COUNT.

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

012410  000463                                  BR      TST63              ;;

012412                                  2$:
012412  104401  012420                          TYPE    ,67$               ;;TYPE ASCIZ STRING
012416  000425                                  BR      66$                ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <200><7><7>#SET ST1 THRESHOLD POT FULLY CLOCKWIZE#
012472                                  66$:
012472  004737  013700                  3$:     JSR     PC,ANYKEY          ;TYPE LAST MESS. AND WAIT FOR OPERATOR.
012476  017746  166674                          MOV     @ASR,-(6) --       ;/SAVE CSR
012502  011637  001424                          MOV     (6),$TMP3          ;/GET CSR.
012506  042737  177707  001424                  BIC     #177707,$TMP3      ;/SAVE RATE BITS.
012514  052737  004005  001424                  BIS     #BIT11!BIT2!BIT0,$TMP3    ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
012522  013777  001424  166646                  MOV     $TMP3,@ASR                ;/LOAD CSR.
                                                                           ;/THIS MUST BE DONE IN
                                                                           ;/ORDER TO XFERR COUNTER
                                                                           ;/TO BUFFER ON ST2.
012530  052777  001000  166640                  BIS     #BIT9,@ASR         ;/GENERATE ON ST2 PULSE
012536  017737  166636  001126                  MOV     @ABR,$BDDAT        ;/READ THE PRESET BUFFER,
                                                                           ;/PREVIOUS COUNTER
012544  012677  166626                          MOV     (6)+,@ASR          ;/CONTENTS ARE IN $BDDAT.
012550  005737  001126                          TST     $BDDAT             ;/RESTORE CSR
012554  001001                                  BNE     TST63              ;;

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

012556  104002                                  ERROR   2                  ;CLOCKWIZE THRESHOLD SETTING OF
                                                                           ;ST1 SHOULD HAVE INHIBITED CLOCK
                                                                           ;FROM COUNTING.
```

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

                                                  ;;*********************************************************************
                                                  ;*TEST 63        *IF ENABLED, CHECK ST1,ST2 IN FROM TESTOR
                                                  ;;*********************************************************************
012560  000004                                    TST63:  SCOPE
012562  012737  000002  001160                            MOV     #2,$TIMES              ;;DO 2 ITERATIONS

012570  005737  001440                                    TST     EXS                    ;OPERATING IN TESTOR MODE?
012574  001510                                            BEQ     ENDP                   ;NO-EXIT THIS TEST.

012576  005077  166574                                    CLR     @ASR                   ;CLEAR THE CSR
012602  012777  177775  166570                            MOV     #-3,@ABR               ;SET TO COUNT THREE TIMES
012610  012777  000061  166560                            MOV     #61,@ASR               ;SET RATE:ST1, MODE1, GO
012616  104401  012624                                    TYPE    ,65$                   ;;TYPE ASCIZ STRING
012622  000433                                            BR      64$                    ;;GET OVER THE ASCIZ
                                                  ;;65$:  .ASCIZ  <200><7><7>#SET ST1 THRESHOLD POT IN THE MIDDLE OF ITS RANGE.#
012712                                            64$:
012712  004737  013700                                    JSR     PC,ANYKEY              ;TYPE LAST MESS WAITE FOR OPERATOR.
012716  005037  001124                                    CLR     $GDDAT                 ;EXPECT COUNTER TO BE CLEAR
012722  017746  166450                                    MOV     @ASR,-(6)              ;/SAVE CSR
012726  011637  001424                                    MOV     (6),$TMP3              ;/GET CSR.
012732  042737  177707  001424                            BIC     #177707,$TMP3          ;/SAVE RATE BITS.
012740  052737  004005  001424                            BIS     #BIT11!BIT2!BIT0,$TMP3   ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
012746  013777  001424  166422                            MOV     $TMP3,@ASR               ;/LOAD CSR.
                                                                                         ;/THIS MUST BE DONE IN
                                                                                         ;/ORDER TO XFERR COUNTER
                                                                                         ;/TO BUFFER ON ST2.
012754  052777  001000  166414                            BIS     #BIT9,@ASR             ;/GENERATE ON ST2 PULSE
012762  017737  166412  001126                            MOV     @ABR,$BDDAT            ;/READ THE PRESET BUFFER,
                                                                                         ;/PREVIOUS COUNTER
012770  012677  166402                                    MOV     (6)+,@ASR              ;/CONTENTS ARE IN $BDDAT.
012774  005737  001126                                    TST     $BDDAT                 ;/RESTORE CSR
                                                                                         ;OF NON-ZERO - REPORT ERROR
013000  001402                                            BEQ     2$

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

013002  104002                                            ERROR   2                      ;INCORRECT # OF ST1'S
                                                                                         ;RECEIVED BY CLOCK

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

013004  000404                                            BR      ENDP

013006                                            2$:
013006  005777  166364                                    TST     @ASR                   ;DID ST2 FLG SET?
013012  100401                                            BMI     ENDP

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

013014  104006                                            ERROR   6                      ;ST2 FLAG DID SET EVEN THOUGH
```

```
                                                                 ;SWITCH WAS TOGGLED.

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


                                                 ;
                                                 ;NOW WE'LL DETERMINE IF WE ARE ALLOWED TO AUTO-SIZE.
                                                 ;IF SO, WE'LL FIND OUT IF THERE ARE OTHER CLOCKS OUT THERE
                                                 ;TO TEST.
                                                 ;

013016  000004                          ENDP:    SCOPE


013020  105737  001215                           TSTB    $ENVM           ;SEE IF APT WILL LET UP AUTO-SIZE.
013024  100537                                   BMI     2$              ;NO - EXIT.


013026  023737  001434  001436                   CMP     MDEVCT,TSTCNT   ;TESTED MAX. UNITS?
013034  001507                                   BEQ     4$              ;YES EXIT.
013036  006337  001426                           ASL     ROTATE          ;POINT NEXT UNIT.
013042  005237  001434                           INC     MDEVCT

013046  062737  000004  001376                   ADD     #4,ASR          ;YES, ADD TO BASE ADDR.
013054  013746  000004                           MOV     ERRVEC,-(6)     ;SAVE CONTENTS OF LOC 4.
013060  012737  013234  000004                   MOV     #1$,ERRVEC      ;SET UP IN CASE NO MORE CLOCKS.

013066  005777  166304                           TST     @ASR            ;TIME OUT HERE IF NO MORE CLOCKS.

                                                                         ;IF HERE, ANOTHER CLOCK FOUND.
013072  005737  001202                           TST     $PASS           ;IS THIS 1ST PASS?
013076  001003                                   BNE     3$              ;NO-GET OUT.
013100  053737  001426  001430                   BIS     ROTATE,UTEST    ;YES-RECORD THIS UNIT.
013106                                  3$:
013106  104401  013114                           TYPE    ,65$            ;;TYPE ASCIZ STRING
013112  000405                                   BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:   .ASCIZ  <15><12>"UNIT #"
013126                                  64$:
013126  013746  001204                           MOV     $DEVCT,-(SP)    ;;SAVE $DEVCT FOR TYPEOUT
013132  104402                                   TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013134  104401  013142                           TYPE    ,67$            ;;TYPE ASCIZ STRING
013140  000406                                   BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:   .ASCIZ  " COMPLETED "
013156                                  66$:
013156  005237  001204                           INC     $DEVCT
013162  104401  013170                           TYPE    ,69$            ;;TYPE ASCIZ STRING
013166  000410                                   BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:   .ASCIZ  " TESTING UNIT #"
013210                                  68$:
013210  013746  001204                           MOV     $DEVCT,-(SP)    ;;SAVE $DEVCT FOR TYPEOUT
013214  104402                                   TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013216  012637  000004                           MOV     (6)+,ERRVEC     ;RESETORE LOC 4.
013222  062737  000010  001402                   ADD     #10,VECT1       ;UPDATE VECTOR ADDR.
013230  000137  002274                           JMP     LOOP            ;TEST NEW UNIT.
```

```
3713
3714  013234                                  1$:
3715  013234  062706  000004                        ADD     #4,SP                   ;/ADD #4 TO STACK POINTER.
3716  013240  012637  000004                        MOV     (6)+,ERRVEC     ;RESTORE LOC 4
3717  013244  022737  000000  001204                CMP     #0,$DEVCT       ;TESTED ONLY ONE UNIT?
3718  013252  001424                                BEQ     2$              ;YES - NO NEED FOR TYPEOUT.
3719
3720  013254                                  4$:
3721  013254  104401  013262                        TYPE    ,71$            ;;TYPE ASCIZ STRING
3722  013260  000405                                BR      70$             ;;GET OVER THE ASCIZ
3723                                          ;;71$:  .ASCIZ  <15><12>"UNIT #"
3724  013274                                  70$:
3725  013274  013746  001204                        MOV     $DEVCT,-(SP)    ;;SAVE $DEVCT FOR TYPEOUT
3726  013300  104402                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3727  013302  104401  013310                        TYPE    ,73$            ;;TYPE ASCIZ STRING
3728  013306  000406                                BR      72$             ;;GET OVER THE ASCIZ
3729                                          ;;73$:  .ASCIZ  " COMPLETED "
3730  013324                                  72$:
3731
3732  013324  013737  001250  001376  2$:    MOV     $BASE,ASR
3733  013332  013737  001244  001402          MOV     $VECT1,VECT1
3734  013340  013737  001204  001442          MOV     $DEVCT,LCNT
3735  013346  005237  001442                  INC     LCNT
3736  013352  012737  000000  001204          MOV     #0,$DEVCT
3737
3738  013360  005037  001434                  CLR     MDEVCT          ;BEGIN TESTING 1ST UNIT.
3739  013364  012737  000001  001426          MOV     #1,ROTATE       ;POINT TO IT.
3740
3741
3742                                          .SBTTL  END OF PASS ROUTINE
3743
3744                                          ;;**************************************************************
3745                                          ;*INCREMENT THE PASS NUMBER ($PASS)
3746                                          ;*IF THERES A MONITOR GO TO IT
3747                                          ;*IF THERE ISN'T JUMP TO LOOP
3748
3749  013372                                  $EOP:
3750  013372  000240                                NOP
3751  013374  005037  001102                        CLR     $TSTNM          ;;ZERO THE TEST NUMBER
3752  013400  005037  001160                        CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
3753  013404  005237  001202                        INC     $PASS           ;;INCREMENT THE PASS NUMBER
3754  013410  042737  100000  001202                BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
3755  013416  005327                                DEC     (PC)+           ;;LOOP?
3756  013420  000001                          $EOPCT: .WORD   1
3757  013422  003122                                BGT     $DOAGN          ;;YES
3758  013424  012737                                MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
3759  013426  000001                          $ENDCT: .WORD   1
3760  013430  013420                                $EOPCT
3761  013432  104401  013440                        TYPE    ,65$            ;;TYPE ASCIZ STRING
3762  013436  000406                                BR      64$             ;;GET OVER THE ASCIZ
3763                                          ;;65$:  .ASCIZ  <15><12>#ENDPASS #
3764  013454                                  64$:
3765  013454  013746  001202                        MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
3766  013460  104402                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
  3767  013462  104401  013470                TYPE    ,67$            ;;TYPE ASCIZ STRING
  3768  013466  000411                       BR      66$             ;;GET OVER THE ASCIZ
  3769                                  ;;67$:  .ASCIZ  #  TOTAL ERRORS  #
  3770  013512                          66$:
  3771  013512  013746  001432                MOV     ERCNT,-(SP)     ;;SAVE ERCNT FOR TYPEOUT
  3772  013516  104402                       TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  3773  013520  104401  013526                TYPE    ,69$            ;;TYPE ASCIZ STRING
  3774  013524  000407                       BR      68$             ;;GET OVER THE ASCIZ
  3775                                  ;;69$:  .ASCIZ  #;  THERE ARE #
  3776  013544                          68$:
  3777  013544  013746  001442                MOV     LCNT,-(SP)      ;;SAVE LCNT FOR TYPEOUT
  3778  013550  104402                       TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  3779  013552  104401  013560                TYPE    ,71$            ;;TYPE ASCIZ STRING
  3780  013556  000411                       BR      70$             ;;GET OVER THE ASCIZ
  3781                                  ;;71$:  .ASCIZ  #  (OCTAL) UNITS.#
  3782  013602                          70$:
  3783  013602  104401  013610                TYPE    ,73$            ;;TYPE ASCIZ STRING
  3784  013606  000415                       BR      72$             ;;GET OVER THE ASCIZ
  3785                                  ;;73$:  .ASCIZ  <200>#THE GOOD UNITS (L TO R) #
  3786  013642                          72$:
  3787  013642  013746  001430                MOV     UTEST,-(SP)     ;;SAVE UTEST FOR TYPEOUT
  3788  013646  104405                       TYPBN                   ;;GO TYPE--BINARY ASCII
  3789  013650  013700  000042        $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
  3790  013654  001405                       BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  3791  013656  000005                       RESET                   ;;CLEAR THE WORLD
  3792  013660  004710                $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  3793  013662  000240                       NOP                     ;;SAVE ROOM
  3794  013664  000240                       NOP                     ;;FOR
  3795  013666  000240                       NOP                     ;;ACT11
  3796  013670                        $DOAGN:
  3797  013670  000137                       JMP     @(PC)+          ;;RETURN
  3798  013672  002274                $RTNAD: .WORD   LOOP
  3799  013674     377     377  000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
  3800          013700                       .EVEN
  3801
  3802                                        ;
  3803                                        ;THIS ROUTINE TYPES LAST MESSAGE AND WAITS FOR AN  OPERATOR
  3804                                        ;RESPONCE.
  3805                                        ;
  3806
  3807  013700  105777  165242        ANYKEY: TSTB    @$TKB           ;CLEAR TTY READY FLAG.
  3808  013704  104401  013712                TYPE    ,65$            ;;TYPE ASCIZ STRING
  3809  013710  000430                       BR      64$             ;;GET OVER THE ASCIZ
  3810                                  ;;65$.  .ASCIZ  <200><7>#SWITCH ST1 3 TIMES,TYPE ANY KEY WHEN DONE..#<7>
  3811  013772                          64$:
  3812
  3813  013772  105777  165146        1$:     TSTB    @$TKS           ;WAIT FOR OPERATOR.
  3814  013776  100375                       BPL     1$
  3815  014000  105777  165142                TSTB    @$TKB           ;CLEAR TTY READY FLAG.
  3816  014004  000207                       RTS PC
  3817
  3818                                        ;
  3819                                  .SBTTL          ;I/O SIGNAL TEST #1 ST1 IN AND ST2 OUT IN AND OUT
  3820                                        ;
```

```
                                ;SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:
                                ;
                                ;       SWITCH  1 - OFF
                                ;               2 - ON
                                ;               3 - OFF
                                ;               4 - OFF
                                ;               5 - ON
                                ;               6 - ON
                                ;               7 - NOT USED
                                ;
                                ; THIS SELECTS TTL THRESHOLDS AND POSITIVE SLOPE FOR
                                ; SCHMITT TRIGGER 1.
                                ;
                                ;PLEASE REMOVE ANY PREVIOUS JUMPER.
                                ;
                                ;JUMPER THE FOLLOWING PINS TOGETHER:
                                ;
                                ;       J1 - SS (ST2 OUT)       TO J1 - VV (ST1-IN)
                                ;
                                ;LOAD AND START AT LOCATION 210
                                ;END PASSES OCCUR IMMEDIATELY AND ARE NOT REPORTED
                                ;ERRORS ARE REPORTED AS IN THE REGULAR LOGIC TEST AND
                                ;THEIR PRINTOUT MAY BE INHIBITED

014006  104407                  IOTST1: CKSWR                   ;CHECK THE SWR
014010  005077  165362          1$:     CLR     @ASR            ;CLEAR THE CSR
014014  005077  165360                  CLR     @ABR            ;CLEAR THE BUFFER REG.
014020  012777  000061  165350          MOV     #61,@ASR        ;RATE ST1, MODE 0, GO.
014026  052777  001000  165342          BIS     #BIT9,@ASR      ;GENERATE A MAINTENANCE ST2.
014034  012777  000005  165334          MOV     #5,@ASR         ;NOW SET TO READ COUNT REG
014042  052777  001000  165326          BIS     #BIT9,@ASR      ;FORCE COUNT -> BUFFER REG.
014050  027727  165324  000001          CMP     @ABR,#1         ;DID COUNT REG ADVANCE ONCE?
014056  001753                          BEQ     IOTST1          ;YES - LOOP.
014060  104000                          ERROR                   ;ST2 OUT TO ST1 IN FAILED.
014062  000751                          BR      IOTST1

                                ;
                                .SBTTL          ;I/O SIGNAL TEST #2 CLOCK OVFLOW OUT TEST.
                                ;

                                ;SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:
                                ;
                                ;       SWITCH  1 - OFF
                                ;               2 - OFF
                                ;               3 - OFF
                                ;               4 - ON
                                ;               5 - ON
                                ;               6 - OFF
                                ;               7 - NOT USED
                                ;
                                ; THIS SELECTS TTL THRESHOLDS AND POSITIVE SLOPE FOR
```

```
                                        ; SCHMITT TRIGGER 2.
                                        ;
                                        ;PLEASE REMOVE ANY PREVIOUS JUMPER.
                                        ;
                                        ;JUMPER THE FOLLOWING PINS TOGETHER:
                                        ;
                                        ;       J1 - RR (CLK OV) TO J1 - TT (ST2-IN)
                                        ;
                                        ;LOAD AND START AT LOCATION 214.
                                        ;END PASSES OCCUR IMMEDIATELY AND ARE NOT REPORTED.
                                        ;ERRORS ARE REPORTED AS IN TH REGULAR LOGIC TEST AND
                                        ;THEIR PRINTOUT MAY BE INHIBITED.
                                        ;
014064  104407              IOTST2: CKSWR                   ;CHECK THE SWR.
014066  005077  165304              CLR     @ASR            ;CLEAR THE CSR.
014072  012777  177777  165300      MOV     #-1,@ABR        ;PRELOAD PRESET BUFFER.
014100  012777  000063  165270      MOV     #63,@ASR        ;RATE ST1, MODE 1, GO.
014106  052777  000400  165262      BIS     #BIT8,@ASR      ;GENERATE A MAIN. ST1.
014114  000240                      NOP
014116  000240                      NOP
014120  005777  165252              TST     @ASR            ;DID OVERFLOW SET ST2 FLAG?
014124  100757                      BMI     IOTST2          ;YES - LOOP
014126  104000                      ERROR                   ;CLK OV OUT TO ST2 IN FAILED.
014130  000755                      BR      IOTST2          ;LOOP

                                        ;
                                .SBTTL  ;I/O SIGNAL TEST #3 ST1 OUT AND ST2 IN
                                        ;

                                        ;
                                        ;SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:
                                        ;       SWITCH  1 - OFF
                                        ;               2 - OFF
                                        ;               3 - OFF
                                        ;               4 - ON
                                        ;               5 - ON
                                        ;               6 - ON
                                        ;               7 - NOT USED
                                        ; THIS SELECTS TTL THRESHOLD AND POSITIVE SLOPE FOR
                                        ; SCHMITT TRIGGER 2.
                                        ;
                                        ;PLEASE REMOVE ANY PREVIOUS JUMPERS.
                                        ;
                                        ;JUMPER THE FOLLOWING PINS TOGETHER:
                                        ;       J1 - UU (ST1 OUT)       TO J1 - TT (ST2-IN)
                                        ;
                                        ;LOAD AND START AT LOCATION 220
                                        ;END PASSES OCCUR IMMEDIATELY AND ARE NOT REPORTED
                                        ;ERRORS ARE REPORTED AS IN THE REGULAR LOGIC TEST AND
                                        ;THEIR PRINTOUT MAY BE INHIBITED
                                        ;
014132  104407              IOTST3: CKSWR                   ;CHECK THE SWR
014134  012777  000001  165234      MOV     #1,@ASR         ;SET GO BIT.
014142  052777  000400  165226      BIS     #BIT8,@ASR      ;GENERATE A MAIN. ST1.
```

```
014150  005777  165222          TST     @ASR            ;DID ST2 FLAG SET?
014154  100401                  BMI     1$
014156  104000                  ERROR                   ;ST1 OUT TO ST2 IN FAILED

014160  032777  010000  165210  1$:     BIT     #BIT12,@ASR     ;DID "FOR" BIT SET?
014166  001761                  BEQ     IOTST3          ;NO - GOOD!
014170  104000                  ERROR                   ;"FOR" BIT SET ON ONLY 1 ST2.
014172  000757                  BR      IOTST3          ;LOOP

                                .SBTTL
                                .SBTTL  *SYSMAC ROUTINES
                                .SBTTL

                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                        ;;**********************************************************************
                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                        ;*      .BYTE   M               ;;M=1 OR 0
                        ;*                                      ;;1=TYPE LEADING ZEROS
                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                        ;*
                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                        ;*$TYPOS OR $TYPOC
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                        ;*
                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPOC                   ;;CALL FOR TYPEOUT

014174  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014200  116637  000001  014417          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014206  112637  014421                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014212  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014216  000406                  BR      $TYPON
014220  112737  000001  014417  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014226  112737  000006  014421          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014234  112737  000005  014416  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014242  010346                  MOV     R3,-(SP)        ;;SAVE R3
014244  010446                  MOV     R4,-(SP)        ;;SAVE R4
014246  010546                  MOV     R5,-(SP)        ;;SAVE R5
014250  113704  014421          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014254  005404                  NEG     R4
014256  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
014262  110437  014420          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014266  113704  014417          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
```

```
 3983  014272  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 3984  014276  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
 3985  014300  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 3986  014302  000404                        BR      3$              ;;GO DO MSB
 3987  014304  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
 3988  014306  006105                        ROL     R5
 3989  014310  006105                        ROL     R5
 3990  014312  010503                        MOV     R5,R3
 3991  014314  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 3992  014316  105337  014420                DECB    $OMODE          ;;TYPE THIS DIGIT?
 3993  014322  100016                        BPL     7$              ;;BR IF NO
 3994  014324  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
 3995  014330  001002                        BNE     4$              ;;TEST FOR 0
 3996  014332  005704                        TST     R4              ;;SUPPRESS THIS 0?
 3997  014334  001403                        BEQ     5$              ;;BR IF YES
 3998  014336  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 3999  014340  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4000  014344  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4001  014350  110337  014414                MOVB    R3,8$           ;;SAVE FOR TYPING
 4002  014354  104401  014414                TYPE    ,8$             ;;GO TYPE THIS DIGIT
 4003  014360  105337  014416        7$:     DECB    $OCNT           ;;COUNT BY 1
 4004  014364  003347                        BGT     2$              ;;BR IF MORE TO DO
 4005  014366  002402                        BLT     6$              ;;BR IF DONE
 4006  014370  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 4007  014372  000744                        BR      2$              ;;GO DO THE LAST DIGIT
 4008  014374  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
 4009  014376  012604                        MOV     (SP)+,R4        ;;RESTORE R4
 4010  014400  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 4011  014402  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 4012  014410  012616                        MOV     (SP)+,(SP)
 4013  014412  000002                        RTI                     ;;RETURN
 4014  014414     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 4015  014415     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 4016  014416     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 4017  014417     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 4018  014420  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 4019                                .SBTTL  BINARY TO ASCII AND TYPE ROUTINE
 4020
 4021                                ;;**************************************************************
 4022                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
 4023                                ;*BINARY-ASCII NUMBER AND TYPE IT.
 4024                                ;*CALL:
 4025                                ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
 4026                                ;*      TYPBN                   ;;TYPE IT
 4027
 4028  014422  010146                $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 4029  014424  016601  000006                MOV     6(SP),R1        ;;GET THE INPUT NUMBER
 4030  014430  000261                        SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
 4031  014432  112737  000060  014474  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
 4032  014440  006101                        ROL     R1              ;;GET THIS BIT
 4033  014442  001406                        BEQ     2$              ;;DONE?
 4034  014444  105537  014474                ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
 4035  014450  104401  014474                TYPE    ,$BIN           ;;GO TYPE THIS BIT
 4036  014454  000241                        CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
```

```
 4037  014456  000765                      BR      1$              ;;GO DO THE NEXT BIT
 4038  014460  012601              2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
 4039  014462  016666  000002  000004      MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4040  014470  012616                      MOV     (SP)+,(SP)
 4041  014472  000002                      RTI                     ;;RETURN TO USER
 4042  014474     000     000      $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
 4043
 4044
 4045                              .SBTTL  ERROR HANDLER ROUTINE
 4046
 4047                              ;;**************************************************************
 4048                              ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 4049                              ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 4050                              ;*AND GO TO $ERRTYP ON ERROR
 4051                              ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 4052                              ;*SW15=1        HALT ON ERROR
 4053                              ;*SW13=1        INHIBIT ERROR TYPEOUTS
 4054                              ;*SW10=1        BELL ON ERROR
 4055                              ;*SW09=1        LOOP ON ERROR
 4056                              ;*CALL
 4057                              ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 4058
 4059  014476                      $ERROR:
 4060  014476  104407                      CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 4061  014500  105237  001103      7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 4062  014504  001775                      BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 4063  014506  013777  001102  164426      MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 4064  014514  032777  002000  164416      BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 4065  014522  001402                      BEQ     1$              ;;NO - SKIP
 4066  014524  104401  001164              TYPE    ,$BELL          ;;RING BELL
 4067  014530  005237  001112      1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 4068  014534  011637  001116              MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 4069  014540  162737  000002  001116      SUB     #2,$ERRPC
 4070  014546  117737  164344  001114      MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 4071  014554  032777  020000  164356      BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 4072  014562  001004                      BNE     20$             ;;SKIP TYPEOUTS
 4073  014564  004737  014704              JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
 4074  014570  104401  001171              TYPE    ,$CRLF
 4075  014574                      20$:
 4076  014574  122737  000001  001214      CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 4077  014602  001007                      BNE     2$              ;;NO,SKIP APT ERROR REPORT
 4078  014604  113737  001114  014616      MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 4079  014612  004737  016376              JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 4080  014616     000              21$:    .BYTE   0
 4081  014617     000                      .BYTE   0
 4082  014620  000777              22$:    BR      22$             ;;APT ERROR LOOP
 4083  014622  005777  164312      2$:     TST     @SWR            ;;HALT ON ERROR
 4084  014626  100002                      BPL     3$              ;;SKIP IF CONTINUE
 4085  014630  000000                      HALT                    ;;HALT ON ERROR!
 4086  014632  104407                      CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 4087  014634  032777  001000  164276  3$: BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 4088  014642  001402                      BEQ     4$              ;;BR IF NO
 4089  014644  013716  001110              MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 4090  014650  005737  001162      4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
```

```
 4091  014654  001402                        BEQ     5$              ;;BR IF NONE
 4092  014656  013716  001162                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 4093  014662                        5$:
 4094
 4095  014662  005237  001432                INC     ERCNT           ;/UPDATE ERROR COUNT.
 4096  014666  001002                        BNE     10$             ;/BUT DON'T LET IT OVERFLOW.
 4097  014670  005337  001432                DEC     ERCNT           ;/KEEP AT 177777 IF OVERFLOW.
 4098  014674                        10$:
 4099  014674  043737  001426  001430        BIC     ROTATE,UTEST    ;/REMOVE UNIT FROM LIST OF GOOD ONES.
 4100  014702  000002                        RTI                     ;/EXIT.
 4101
 4102                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 4103
 4104                                  ;;**************************************************************
 4105                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 4106                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 4107                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 4108
 4109  014704                        $ERRTYP:
 4110  014704  104401  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 4111  014710  010046                        MOV     R0,-(SP)        ;;SAVE R0
 4112  014712  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
 4113  014714  153700  001114                BISB    @#$ITEMB,R0
 4114  014720  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
 4115                                                                ;;TYPE THE PC OF THE ERROR
 4116  014722  013746  001116                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 4117                                                                ;;ERROR ADDRESS
 4118  014726  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4119  014730  000426                        BR      6$              ;;GET OUT
 4120  014732  005300                1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 4121  014734  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
 4122  014736  006300                        ASL     R0
 4123  014740  006300                        ASL     R0
 4124  014742  062700  001256                ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 4125  014746  012037  014756                MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
 4126  014752  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 4127  014754  104401                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
 4128  014756  000000                2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
 4129  014760  104401  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 4130  014764  012037  014774        3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
 4131  014770  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
 4132  014772  104401                        TYPE                    ;;TYPE THE "DATA HEADER"
 4133  014774  000000                4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
 4134  014776  104401  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 4135  015002  011000                5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
 4136  015004  001004                        BNE     7$              ;;GO TYPE THE DATA
 4137  015006  012600                6$:     MOV     (SP)+,R0        ;;RESTORE R0
 4138  015010  104401  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 4139  015014  000207                        RTS     PC              ;;RETURN
 4140  015016                        7$:
 4141  015016  013046                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
 4142  015020  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4143  015022  005710                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
 4144  015024  001770                        BEQ     6$              ;;BR IF NO
```

```
4145  015026  104401  015034                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
4146  015032  000771                          BR      7$              ;;LOOP
4147  015034  020040      000       8$:       .ASCIZ  /  /            ;;TWO(2) SPACES
4148          015040                          .EVEN
4149                                  .SBTTL  SCOPE HANDLER ROUTINE
4150
4151                                  ;;**********************************************************************
4152                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4153                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4154                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4155                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4156                                  ;*SW14=1         LOOP ON TEST
4157                                  ;*SW11=1         INHIBIT ITERATIONS
4158                                  ;*SW09=1         LOOP ON ERROR
4159                                  ;*SW08=1         LOOP ON TEST IN SWR<7:0>
4160                                  ;*CALL
4161                                  ;*      SCOPE           ;;SCOPE=IOT
4162
4163  015040                          $SCOPE:
4164  015040  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
4165  015042  104407                          CKSWR
4166  015044  032777  040000  164066  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
4167  015052  001114                          BNE     $OVER           ;;YES IF SW14=1
4168                                  ;*****START OF CODE FOR THE XOR TESTER*****
4169  015054  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
4170                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
4171  015056  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
4172  015062  012737  015102  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
4173  015070  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
4174  015074  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
4175  015100  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
4176  015102  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
4177  015104  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
4178  015110  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
4179  015112                          6$:;*****END OF CODE FOR THE XOR TESTER*****
4180  015112  032777  000400  164020          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
4181  015120  001404                          BEQ     2$              ;;BR IF NO
4182  015122  127737  164012  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
4183  015130  001465                          BEQ     $OVER           ;;BR IF YES
4184  015132  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
4185  015136  001421                          BEQ     3$              ;;BR IF NO
4186  015140  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
4187  015146  101015                          BHI     3$              ;;BR IF NO
4188  015150  032777  001000  163762          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
4189  015156  001404                          BEQ     4$              ;;BR IF NO
4190  015160  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
4191  015166  000446                          BR      $OVER
4192  015170  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
4193  015174  005037  001160                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
4194  015200  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
4195  015202  032777  004000  163730  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
4196  015210  001011                          BNE     1$              ;;BR IF YES
4197  015212  005737  001202                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
4198  015216  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
```

```
4199  015220  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
4200  015224  023737  001160  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
4201  015232  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
4202  015234  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
4203  015242  013737  015320  001160          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
4204  015250  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
4205  015254  113737  001102  001200          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
4206  015262  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
4207  015266  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
4208  015272  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
4209  015276  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4210  015304  013777  001102  163630  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
4211  015312  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
4212  015316  000002                          RTI                     ;;FIXES PS
4213  015320  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
4214                                   .SBTTL  TTY INPUT ROUTINE
4215
4216                                   ;;**************************************************************
4217                                   .ENABL  LSB
4218
4219                                   ;;**************************************************************
4220                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4221                                   ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
4222                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
4223                                   ;*WHEN OPERATING IN TTY FLAG MODE.
4224  015322  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
4225  015330  001074                          BNE     15$             ;;BRANCH IF NO
4226  015332  105777  163606                  TSTB    @$TKS           ;;CHAR THERE?
4227  015336  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
4228  015340  117746  163602                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
4229  015344  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
4230  015350  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
4231  015354  001062                          BNE     15$             ;;NO, RETURN TO USER
4232  015356  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
4233  015364  001456                          BEQ     15$             ;;BRANCH IF YES
4234
4235  015366  104401  016047                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
4236  015372  104401  016054          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
4237  015376  013746  000176                  MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
4238  015402  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4239  015404  104401  016065                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
4240  015410  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
4241  015412  005046                          CLR     -(SP)           ;;THE NEW SWR
4242  015414  105777  163524          7$:     TSTB    @$TKS           ;;CHAR THERE?
4243  015420  100375                          BPL     7$              ;;IF NOT TRY AGAIN
4244
4245  015422  117746  163520                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
4246  015426  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
4247
4248
4249
4250  015432  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
4251  015436  001005                          BNE     10$             ;;BRANCH IF NOT
4252  015440  104401  016042                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
```

```
4253  015444  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
4254  015450  000757                          BR      19$             ;;LET'S TRY IT AGAIN
4255
4256
4257  015452  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
4258  015456  001022                          BNE     16$             ;;BRANCH IF NO
4259  015460  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
4260  015464  001403                          BEQ     11$             ;;BRANCH IF YES
4261  015466  016677  000002  163444          MOV     2(SP),@SWR      ;;SAVE NEW SWR
4262  015474  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
4263  015500  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
4264  015504  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
4265  015512  001003                          BNE     15$             ;;BRANCH IF NOT
4266  015514  012777  000100  163422          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
4267  015522  000002          15$:    RTI                             ;;RETURN
4268  015524  004737  016310          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
4269  015530  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
4270  015534  002420                          BLT     18$             ;;BRANCH IF YES
4271  015536  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
4272  015542  003015                          BGT     18$             ;;BRANCH IF YES
4273  015544  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
4274  015550  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
4275  015554  001403                          BEQ     17$             ;;BRANCH IF YES
4276  015556  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
4277  015560  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
4278  015562  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
4279  015564  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
4280  015570  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
4281  015574  000707                          BR      7$              ;;GET THE NEXT ONE
4282  015576  104401  001170          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
4283  015602  000720                          BR      20$             ;;SIMULATE CONTROL-U
4284                                .DSABL  LSB
4285
4286
4287                                ;;**************************************************************
4288                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4289                                ;*CALL:
4290                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
4291                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
4292                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
4293                                ;
4294
4295  015604  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
4296  015606  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
4297  015614  105777  163324          1$:     TSTB    @$TKS           ;;WAIT FOR
4298  015620  100375                          BPL     1$              ;;A CHARACTER
4299  015622  117766  163320  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
4300  015630  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
4301  015636  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
4302  015644  001013                          BNE     3$              ;;BRANCH IF NO
4303  015646  105777  163272          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
4304  015652  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
4305  015654  117746  163266                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
4306  015660  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
```

```
 4307  015664  022627  000021              CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4308  015670  001366                      BNE     2$              ;;IF NOT DISCARD IT
 4309  015672  000750                      BR      1$              ;;YES, RESUME
 4310  015674  026627  000004  000140  3$: CMP     4(SP),#140      ;;IS IT UPPER CASE?
 4311  015702  002407                      BLT     4$              ;;BRANCH IF YES
 4312  015704  026627  000004  000175      CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 4313  015712  003003                      BGT     4$              ;;BRANCH IF YES
 4314  015714  042766  000040  000004      BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 4315  015722  000002                  4$: RTI                     ;;GO BACK TO USER
 4316                                  ;;**************************************************
 4317                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 4318                                  ;*CALL:
 4319                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 4320                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 4321                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 4322
 4323  015724  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 4324  015726  012703  016032          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 4325  015732  022703  016042          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
 4326  015736  101405                          BLOS    4$              ;;BR IF YES
 4327  015740  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4328  015742  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4329  015744  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4330  015750  001003                          BNE     3$              ;;SKIP IF NOT
 4331  015752  104401  001170          4$:     TYPE    $QUES           ;;TYPE A '?'
 4332  015756  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4333  015760  111337  016030          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4334  015764  104401  016030                  TYPE    ,9$
 4335  015770  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4336  015774  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 4337  015776  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4338  016002  104401  001172                  TYPE    $LF             ;;TYPE A LINE FEED
 4339  016006  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4340  016010  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4341  016012  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4342  016020  012766  016032  000004          MOV     #$TTYIN,4(SP)
 4343  016026  000002                          RTI                     ;;RETURN
 4344  016030     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 4345  016031     000                          .BYTE   0               ;;TERMINATOR
 4346  016032  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 4347  016042  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 4348  016047     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 4349  016054  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 4350  016062  020075     000
 4351  016065     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 4352  016072  036440  000040
 4353                                  .SBTTL  TYPE ROUTINE
 4354
 4355                                  ;;**************************************************************
 4356                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 4357                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 4358                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 4359                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 4360                                  ;*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
                                        ;*
016076  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
016102  100002                          BPL     1$              ;;BR IF YES
016104  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
016106  000430                          BR      3$              ;;LEAVE
016110  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
016112  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
016116  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
016124  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
016126  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
016134  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
016136  010037  016146                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
016142  004737  016366                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
016146  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
016150  132737  000040  001215  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
016156  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
016160  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
016162  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
016164  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
016166  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
016170  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
016174  000002                          RTI                     ;;RETURN
016176  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
016202  001430                          BEQ     8$
016204  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
016210  001006                          BNE     5$
016212  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
016214  104401                          TYPE                    ;;TYPE A CR AND LF
016216  001171                          $CRLF
016220  105037  016354                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
016224  000755                          BR      2$              ;;GET NEXT CHARACTER
016226  004737  016310          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
016232  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
016236  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
016240  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
016244  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
016250  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
016252  004737  016310                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
016256  105337  016354                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
016262  000770                          BR      7$              ;;LOOP

                                        ;HORIZONTAL TAB PROCESSOR

016264  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
016270  004737  016310          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
```

```
4415  016274  132737  000007  016354          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4416  016302  001372                          BNE     9$              ;;TAB STOP
4417  016304  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
4418  016306  000724                          BR      2$              ;;GET NEXT CHARACTER
4419  016310  105777  162634          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4420  016314  100375                          BPL     $TYPEC
4421  016316  116677  000002  162626          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4422  016324  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
4423  016332  001003                          BNE     1$              ;;BRANCH IF NO
4424  016334  105037  016354                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
4425  016340  000406                          BR      $TYPEX          ;;EXIT
4426  016342  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
4427  016350  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
4428  016352  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
4429  016354  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
4430  016356  000207          $TYPEX: RTS     PC
4431
4432                          .SBTTL  APT COMMUNICATIONS ROUTINE
4433
4434                          ;;**************************************************************
4435  016360  112737  000001  016624  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
4436  016366  112737  000001  016622  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
4437  016374  000403                          BR      $ATYC
4438  016376  112737  000001  016624  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
4439  016404          $ATYC:
4440  016404  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
4441  016406  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
4442  016410  105737  016622                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
4443  016414  001450                          BEQ     5$              ;;IF NOT:  BR
4444  016416  122737  000001  001214          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
4445  016424  001031                          BNE     3$              ;;IF NOT:  BR
4446  016426  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
4447  016434  001425                          BEQ     3$              ;;IF NOT:  BR
4448  016436  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
4449  016442  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
4450  016450  005737  001174          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
4451  016454  001375                          BNE     1$              ;;IF NOT:  WAIT
4452  016456  010037  001210                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
4453  016462  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
4454  016464  001376                          BNE     2$
4455  016466  163700  001210                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
4456  016472  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
4457  016474  010037  001212                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
4458  016500  012737  000004  001174          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
4459  016506  000413                          BR      5$
4460  016510  017637  000004  016534  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
4461  016516  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
4462  016524  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
4463  016530  004737  016076                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
4464  016534  000000          4$:     .WORD   0
4465  016536          5$:
4466  016536  105737  016624          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
4467  016542  001416                          BEQ     12$             ;;IF NOT:  BR
4468  016544  005737  001214                  TST     $ENV            ;;RUNNING UNDER APT?
```

```
4469  016550  001413                        BEQ     12$             ;;IF NOT:  BR
4470  016552  005737  001174          11$:  TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
4471  016556  001375                        BNE     11$             ;;IF NOT:  WAIT
4472  016560  017637  000004  001176        MOV     @4(SP),$FATAL   ;;GET ERROR #
4473  016566  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
4474  016574  005237  001174                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
4475  016600  105037  016624          12$:  CLRB    $FFLG           ;;CLEAR FATAL FLAG
4476  016604  105037  016623                CLRB    $LFLG           ;;CLEAR LOG FLAG
4477  016610  105037  016622                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
4478  016614  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
4479  016616  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
4480  016620  000207                        RTS     PC              ;;RETURN
4481  016622     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
4482  016623     000          $LFLG:  .BYTE   0               ;;LOG FLAG
4483  016624     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
4484        016626                        .EVEN
4485        000200                APTSIZE=200
4486        000001                APTENV=001
4487        000100                APTSPOOL=100
4488        000040                APTCSUP=040
4489                              .SBTTL  POWER DOWN AND UP ROUTINES
4490
4491                              ;;*****************************************************************
4492                              ;POWER DOWN ROUTINE
4493  016626  012737  016766  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
4494  016634  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
4495  016642  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
4496  016644  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
4497  016646  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
4498  016650  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
4499  016652  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
4500  016654  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
4501  016656  017746  162256                MOV     @$WR,-(SP)      ;;PUSH @SWR ON STACK
4502  016662  010637  016772                MOV     SP,$SAVR6       ;;SAVE SP
4503  016666  012737  016700  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
4504  016674  000000                        HALT
4505  016676  000776                        BR      .-2             ;;HANG UP
4506
4507                              ;;*****************************************************************
4508                              ;POWER UP ROUTINE
4509  016700  012737  016766  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
4510  016706  013706  016772                MOV     $SAVR6,SP       ;;GET SP
4511  016712  005037  016772                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
4512  016716  005237  016772          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
4513  016722  001375                        BNE     1$              ;;OF  WORD
4514  016724  012677  162210                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
4515  016730  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
4516  016732  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
4517  016734  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
4518  016736  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
4519  016740  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
4520  016742  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
4521  016744  012737  016626  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
4522  016752  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
```

```
4523  016760  104401                          TYPE                    ;REPORT THE POWER FAILURE
4524  016762  016774              $PWRMG:     .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
4525  016764  000002                          RTI
4526  016766  000000              $ILLUP:     HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
4527  016770  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
4528  016772  000000              $SAVR6:     0                       ;;PUT THE SP HERE
4529  016774  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
4530  017002  000122
4531                                          .EVEN
4532                                          ;*
4533                                          ;*THIS ROUTINE WILL PROTECT THE PROGRAM
4534                                          ;*FROM INTERUPTS (BAD ONES).
4535                                          ;*
4536                                          ;*THE TRAP CATCHER IS SET UP FOR
4537                                          ;*      .WORD   .+2
4538                                          ;*      JSR PC,R0
4539                                          ;*
4540                                          ;*ILLEGAL INTERRUPTS OR INTERRUPTS TO THE WRONG VECTOR
4541                                          ;*GOTO THE VECTOR AND PCITK UP THE ".+2" AS AN ADDRESS
4542
4543                                          ;*AND "4700" AS NEW STATUS.
4544                                          ;*THE .+2 AS A PC WILL CAUSE EXECUTION OF THE "JSR PC,R0" (AN ILLEGAL INSTR.).
4545                                          ;*AND TRAP TO LOCATION "4". IN LOCATION 4 WE HAVE A
4546                                          ;*POINTER HERE. IF THIS CONDITION CAUSES A TRAP TO LOC. 4.
4547                                          ;*WE WILL REPORT IT IN THE SAME MANNER THAT WER WOULD
4548                                          ;*REPORT ANY OTHER ERROR.
4549                                          ;*IF A BUSS ERROR TRAP DID OCCUT AND CAUSE A TRAP TO 4.
4550                                          ;*WE WILL HALT.
4551
4552  017004  011637  017050      IOTRD:      MOV     (6),TRTO        ;GET WHERE WE CAME TO.
4553  017010  162737  000004  017050          SUB     #4,TRTO         ;FORM READ ADDR.
4554
4555  017016  023727  017050  001000          CMP     TRTO,#1000      ;DID TRAP FROM LESS THAN ADDR. 1000?
4556  017024  003402                          BLE     2$              ;NO-CONTINUE.
4557
4558  017026  000000              1$:         HALT                    ;A BUSS ERROR TIME OUT TRAP BROUGHT US HERE.
4559                                                                  ;ADDRESS CONTAINED IN TRTO.
4560
4561  017030  000776                          BR      1$              ;DON'T ALLOW CONTINUE.
4562
4563  017032  016637  000004  017052  2$:     MOV     4(6),TRFRO      ;GET TRAPPED FROM ADDR.
4564  017040  062706  000004                  ADD     #4,SP                   ;/ADD #4 TO STACK POINTER.
4565

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

4568  017044  104004                          ERROR   4               ;ERROR! ILLEGAL INTERRUPT OR
4569                                                                  ;INTERRUPT TO WRONG VECTOR.
4570                                                                  ;IF TEST NO. IS LESS THAN 10,ITS
4571                                                                  ;LIKELY(BUT NO EXCLUSIVELY)TO BE A
4572                                                                  ;DEVICE OTHER THAN THE DEVICE UNDER TEST.
4573                                                                  ;IF THE INTERRUPT OCCURED
4574                                                                  ;DURING AN INTERRUPT TEST, I'D
4575                                                                  ;SUSPECT A PROBLEM WITH THE DEVICE UNDER TEST.
4576                                                                  ;IF THE ADDRESS THE INTERRUPT
```

```
4577                                                    ;VECTORED TO IS WITHIN THE RANGE OF
4578                                                    ;VECTORS ASSIGNED TO THE DEVICE,
4579                                                    ;THEN I'D SUSPECT THE DEVICE
4580                                                    ;INTERRUPTD ILLEGALLY.
4581                                                    ;IF THE ADDRESS THE INTERRUPT
4582                                                    ;VECTORED TO IS OUTSIDE OF THE
4583                                                    ;RANGE ASSIGNED TO THE DEVICE
4584                                                    ;I'D SUSPECT THAT THE
4585                                                    ;DEVICE PUT THE WRONG INTERRUPT
4586                                                    ;VICTOR ON THE BUS DURING THE INTERRUPT
4587                                                    ;PROCESS.
4588                                                    ;       NOTE:
4589                                                    ;FOR THIS ERROR - DON'T USE
4590                                                    ; "LOOP ON ERROR" OPTION.
4591                                                    ;ALSO EXPECT THAT THE INTERRUPT TEST TO
4592                                                    ;WILL REPOT THAT THE DEVICE DIDN'T
4593                                                    ;INTERRUPT.
4594                                                    ;FOLLOW THE RECOMMENDED PROCEEDURE
4595                                                    ; IN THE DOCUMENT (ON THIS DIAGNOSTIC)
4596                                                    ;FOR LOOPING ON TEST.
4597
4598

                                        ;;SSSSSSSSSSSSSSSSSSSSSS››› ERROR ‹‹‹SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

4601  017046  000002                            RTI
4602  017050  000000                    TRTO:   .WORD   0               ;CONTAINS ADDR. WE TRAPPED OR INTERRUPTED TO.
4603  017052  000000                    TRFRO:  .WORD   0               ;CONTAINS ADDR. WE TRAPPED OR INTR. FROM.
4604                                    .SBTTL  TRAP DECODER
4605
4606                                    ;;**************************************************************
4607                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4608                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4609                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4610                                    ;*GO TO THAT ROUTINE.
4611
4612  017054  010046                    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
4613  017056  016600  000002                    MOV     2(SP),R0        ;;GET TRAP ADDRESS
4614  017062  005740                            TST     -(R0)           ;;BACKUP BY 2
4615  017064  111000                            MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
4616  017066  006300                            ASL     R0              ;;POSITION FOR INDEXING
4617  017070  016000  017110                    MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
4618  017074  000200                            RTS     R0              ;;GO TO ROUTINE
4619
4620
4621                                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
4622
4623  017076  011646                    $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
4624  017100  016666  000004  000002            MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
4625  017106  000002                            RTI                     ;;RESTORE THE PSW
4626
4627                                    .SBTTL  TRAP TABLE
4628
4629                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
4630                                    ;*BY THE "TRAP" INSTRUCTION.
```

```
                                                ;       ROUTINE
                                                ;       -------
017110  017076                                  $TRPAD: .WORD   $TRAP2
017112  016076                                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
017114  014220                                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
017116  014174                                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
017120  014234                                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
017122  014422                                          $TYPBN  ;;CALL=TYPBN    TRAP+5(104405)  TYPE BINARY (ASCII) NUMBER

017124  015372                                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

017126  015322                                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
017130  015604                                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
017132  015724                                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
017134  005015  046103  041517                  EM1:    .ASCIZ  <15><12>/CLOCK SR FUNCTION ERROR/
017142  020113  051123  043040
017150  047125  052103  047511
017156  020116  051105  047522
017164  000122
017166  005015  046103  041517                  EM2:    .ASCIZ  <15><12>/CLOCK SR DATA ERROR/
017174  020113  051123  042040
017202  052101  020101  051105
017210  047522  000122
017214  005015  046103  041517                  EM3:    .ASCIZ  <15><12>/CLOCK BR DATA ERROR/
017222  020113  051102  042040
017230  052101  020101  051105
017236  047522  000122
017242  044600  052116  051105                  EM4:    .ASCIZ  <200>/INTERRUPT ERROR/
017250  052522  052120  042440
017256  051122  051117     000
017263     015  041412  052517                  EM5:    .ASCIZ  <15><12>/COUNT REG. ERROR/
017270  052116  051040  043505
017276  020056  051105  047522
017304  000122
017306  005015  047503  047125                  EM11:   .ASCIZ  <15><12>*COUNT ERROR *
017314  020124  051105  047522
017322  020122     000
017325     015  041412  052517                  EM12:   .ASCIZ  <15><12>*COUNT FUNCTION ERROR*
017332  052116  043040  047125
017340  052103  047511  020116
017346  051105  047522  000122
017354  005015  046103  041517                  EM16:   .ASCIZ  <15><12>*CLOCK INTERRUPT ERROR *
017362  020113  047111  042524
017370  051122  050125  020124
017376  051105  047522  020122
017404     000
017405     015  051012  050105                  EM20:   .ASCIZ  <15><12>*REPEATABILITY ERROR *
017412  040505  040524  044502
017420  044514  054524  042440
017426  051122  051117  000040
017434  005015  042101  051104                  EM26:   .ASCIZ  <15><12>*ADDRESSING ERROR*
017442  051505  044523  043516
017450  042440  051122  051117
```

```
4685  017456     000
4686
4687  017457     015  042412  051122  DH1:    .ASCIZ  <15><12>#ERRPC  ASR     WAS     S/B#
4688  017464  041520  040411  051123
4689  017472  053411  051501  051411
4690  017500  041057     000
4691  017503     015  042412  051122  DH3:    .ASCIZ  <15><12>#ERRPC  ABR     WAS     S/B#
4692  017510  041520  040411  051102
4693  017516  053411  051501  051411
4694  017524  041057     000
4695  017527     200  051105  050122  DH4A:   .ASCIZ  <200>#ERRPC   TO      FROM ADDR.#
4696  017534  020103  020040  047524
4697  017542  020040  020040  020040
4698  017550  051106  046517  040440
4699  017556  042104  027122     000
4700  017563     015  042412  051122  DH12:   .ASCIZ  <15><12>#ERRPC  ASR     #
4701  017570  041520  040411  051123
4702  017576  000011
4703  017600  005015  051105  050122  DH20:   .ASCIZ  <15><12>#ERRPC  ASR     2NDCNT  1STNCT  3RDCNT#
4704  017606  004503  051501  004522
4705  017614  047062  041504  052116
4706  017622  030411  052123  041516
4707  017630  004524  051063  041504
4708  017636  052116     000
4709  017641     015  042412  051122  DH26:   .ASCIZ  <15><12>#ERRPC  CLOCK ADDR.#
4710  017646  041520  041411  047514
4711  017654  045503  040440  042104
4712  017662  027122     000
4713
4714          017666                          .EVEN
4715
4716  017666  001116  001376  001126  DT1:    .WORD   $ERRPC,ASR,$BDDAT,$GDDAT,0
4717  017674  001124  000000
4718  017700  001116  001400  001126  DT3:    .WORD   $ERRPC,ABR,$BDDAT,$GDDAT,0
4719  017706  001124  000000
4720  017712  001116  017050  017052  DT4:    .WORD   $ERRPC,TRTO,TRFRO,0
4721  017720  000000
4722  017722  001116  001376  000000  DT12:   .WORD   $ERRPC,ASR,0
4723  017730  001116  001376  001126  DT20:   .WORD   $ERRPC,ASR,$BDDAT,$GDDAT,$TMP0,0
4724  017736  001124  001420  000000
4725  017744  001116  001376  001126  DT22:   .WORD   $ERRPC,ASR,$BDDAT,$TMP0,0
4726  017752  001420  000000
4727  017756  001116  001420  000000  DT26:   .WORD   $ERRPC,$TMP0,0
4728
4729  017764  000000  000000          DF0:    .WORD   0,0
4730
4731          000001                          .END
```

```
BIT12 = 010000   795* 2017 2061 2589 2805 2809 2834 2865 3933
BIT13 = 020000   794* 1333 1334 1348 2103 4071
BIT14 = 040000   793* 1288 1289 1303 2765 4166
BIT15 = 100000   792* 2809
BIT2  = 000004   815* 1603 1604 1618 1911 1956 2086 2137 2196 2346 2397 2448 2499
                2550 2605 2653 2895 2936 2983 3019 3067 3103 3151 3187 3236 3272
                3323 3362 3437 3500 3557 3588 3630
BIT3  = 000010   814* 1558 1559 1573 2888 2924
BIT4  = 000020   813* 1513 1514 1528 2131 2175 2240 2279
BIT5  = 000040   812* 1468 1469 1483 2131 2175 2234 2240 2279
BIT6  = 000100   811* 1423 1424 1438
BIT7  = 000200   810*
BIT8  = 000400   809* 2132 2177 2242 2281 2305 2644 2726 3892 3928
BIT9  = 001000   808* 1916 1961 2069 2104 2142 2201 2351 2402 2453 2504 2555 2610
                2658 2766 2802 2803 2854 2855 2900 2941 2988 3024 3072 3108 3156
                3192 3241 3277 3328 3367 3418 3442 3481 3505 3562 3593 3635 3851
                3853
BPTVEC= 000014   824*
CKSWR = 104407  3847 3888 3926 4060 4086 4164 4165 4643*
CR    = 000015   732* 4422 4432
CRLF  = 000200   733* 1167 4393 4432
DDISP = 177570   739*  901 1121
DFO     017764   988  996 1004 1012 1020 1028 1036 1044 1052 1060 4729*
DH1     017457   986  994 1018 1050 4687*
DH12    017563  1026 1034 4700*
DH20    017600  1042 4703*
DH26    017641  1058 4709*
DH3     017503  1002 4691*
DH4A    017527  1010 4695*
DISPLA  001142   901* 1121* 1129* 4063* 4210*
DISPRE  000174   700* 1129
DR      001414  1070* 2018 2086 2181
DR2     001416  1071* 2017* 2061*
DSWR  = 177570   738*  900 1120
DT1     017666   987  995 1019 4716*
DT12    017722  1027 1035 4722*
DT20    017730  1043 4723*
DT22    017744  1051 4725*
DT26    017756  1059 4727*
DT3     017700  1003 4718*
DT4     017712  1011 4720*
EMTVEC= 000030   827* 1105* 1106*
EMI     017134   985 4646*
EM11    017306  1049 4666*
EM12    017325  1025 4669*
EM16    017354  1033 4673*
EM2     017166   993 4651*
EM20    017405  1041 4678*
EM26    017434  1057 4682*
EM3     017214  1001 4655*
EM4     017242  1009 4659*
EM5     017263  1017 4662*
ENDP    013016  3542 3616 3650 3654 3670*
ERCNT   001432  1077* 1181* 3771 4095* 4097*
```

```
ERRVEC= 000004          820*  1118   1119*  1130*  1145*  1211   1212*  1236*  1245   1246*  1270*  3683   3684*
                       3710*  3716*  4171   4172*  4174*  4177*
EXS     001440         1080*  1084*  1089*  1093*  1169   2015   2059   2084   2179   2229   2253   3540   3615
GNS   = ****** U       1166   1173   1178   3550   3582   3623   3695   3701   3706   3723   3729   3763   3769
                       3775   3781   3785   3810   4635   4636   4637   4638   4639   4641   4643   4644   4645
GTSWR = 104406         1161   4641*
HT    = 000011          730*  4391   4432
IOTRD   017004          698   1145   4552*
IOTST1  014006          710   3847*  3855   3857
IOTST2  014064          711   3888*  3896   3898
IOTST3  014132          712   3926*  3934   3936
IOTVEC= 000020          825*  1103*  1104*
LCNT    001442         1081*  3734*  3735*  3777
LF    = 000012          731*  4426   4432
LOOP    002274         1185*  3712   3798
MDEVCT  001434         1078*  1180*  3677   3680*  3738*
PC    =%000007          751*  3552*  3584*  3625*  3755*  3758*  3792*  3797   3816*  4073*  4079*  4139*  4268*
                       4381*  4400*  4407*  4414*  4428*  4430*  4463*  4480*
PIRQ  = 177772           737*
PIRQVE= 000240           831*
PRIOR   001412         1069*
PR0   = 000000           754*
PR1   = 000040           755*
PR2   = 000100           756*
PR3   = 000140           757*
PR4   = 000200           758*
PR5   = 000240           759*
PR6   = 000300           760*
PR7   = 000340           761*
PS    = 177776           734*   735
PSW   = 177776           735*
PWRVEC= 000024           826*  1109*  1110*  4493*  4494*  4503*  4509*  4521*  4522*
RDCHR = 104410         4327    4644*
RDLIN = 104411         4645*
RESVEC= 000010           821*
ROTATE  001426         1075*  1183*  1184   3679*  3691   3739*  4099
RSTART  002104          709   1168*
R0    =%000000          742*  2235*  2258*  2338*  2340*  2389*  2391*  2440*  2442*  2491*  2493*  2542*  2544*
                       2597*  2599*  2645*  2647*  2687*  2688*  2829*  2830*  2889*  2890*  2925*  2930*  2972*
                       2975*  3008*  3012*  3056*  3059*  3092*  3096*  3140*  3143*  3176*  3180*  3225*  3228*
                       3261*  3265*  3310*  3313*  3349*  3353*  3413*  3415*  3476*  3478*  3789*  3792   4111
                       4112*  4113*  4120*  4121*  4122*  4123*  4124*  4125   4130   4135*  4137*  4141   4143
                       4374   4375*  4380   4385   4388*  4440   4448*  4452   4453   4455*  4456*  4457   4479*
                       4495   4520*  4612   4613*  4614   4615*  4616*  4617*  4618*
R1    =%000001          743*  3308*  3316*  3348*  3355*  4028   4029*  4032*  4038*  4441   4478*  4496   4519*
R2    =%000002          744*  4497   4518*
R3    =%000003          745*  3975   3984*  3990*  3991*  3994*  3999*  4000*  4001   4010*  4323   4324*  4325
                       4328*  4329   4333   4335   4337*  4339*  4498   4517*
R4    =%000004          746*  3976   3978*  3979*  3980*  3981   3982*  3996   3998*  4006*  4009*  4499   4516*
R5    =%000005          747*  3977   3983*  3985*  3987*  3988*  3989*  3990   4008*  4500   4515*
R6    =%000006          748*  1097*  1098*  1099
R7    =%000007          749*
SP    =%000006          750*  1101*  1118*  1126*  1130   1139*  1140*  1211*  1220*  1236   1245*  1254*  1270
                       2717*  2718*  2729*  2730*  2736*  2737*  2749*  2758*  2759*  2768*  2769*  2775*  2776*
```

```
                    2788*  3697*  3708*  3715*  3725*  3765*  3771*  3777*  3787*  3967*  3968   3969   3970*
                    3975*  3976*  3977*  3983   4008   4009   4010   4011*  4012*  4028*  4029   4038   4039*
                    4040*  4068   4089*  4092*  4111*  4116*  4137   4141*  4171*  4174   4176   4177   4206
                    4207   4211*  4228*  4229*  4230   4237*  4240*  4241*  4245*  4246*  4250   4253*  4257
                    4259   4261   4262*  4269   4271   4273*  4274   4276*  4277*  4278*  4279*  4280*  4295*
                    4296*  4299*  4300*  4301   4305*  4306*  4307   4310   4312   4314*  4323*  4328   4339
                    4340*  4341*  4342*  4374*  4375   4385*  4387   4388   4389*  4391   4393   4395   4401
                    4403*  4405*  4413*  4417   4421   4422   4426   4440*  4441*  4448   4449*  4460   4461*
                    4462*  4472   4473*  4478   4479   4495*  4496*  4497*  4498*  4499*  4500*  4501*  4502
                    4510*  4514   4515   4516   4517   4518   4519   4520   4564*  4612*  4613   4623*  4624*
STACK = 001100       725*  1101
START = 001474       708   1091*
STKLMT= 177774       736*
SWR     001140       900*  1099   1120*  1122   1128*  1135*  1159   2589   4064   4071   4083   4087   4166
                    4180   4182   4188   4195   4224   4261*  4501   4514*
SWREG   000176       701*  1128   1159   4224   4237
SW0   = 000001       789*
SW00  = 000001       779*   789
SW01  = 000002       778*   788
SW02  = 000004       777*   787
SW03  = 000010       776*   786
SW04  = 000020       775*   785
SW05  = 000040       774*   784
SW06  = 000100       773*   783
SW07  = 000200       772*   782
SW08  = 000400       771*   781
SW09  = 001000       770*   780
SW1   = 000002       788*
SW10  = 002000       769*
SW11  = 004000       768*
SW12  = 010000       767*
SW13  = 020000       766*
SW14  = 040000       765*
SW15  = 100000       764*
SW2   = 000004       787*
SW3   = 000010       786*
SW4   = 000020       785*
SW5   = 000040       784*
SW6   = 000100       783*
SW7   = 000200       782*
SW8   = 000400       781*
SW9   = 001000       780*
TBITVE= 000014       822*
TKVEC = 000060       829*
TPVEC = 000064       830*
TRAPVE= 000034       828*  1107*  1108*
TRFRO   017052      4563*  4603*  4720
TRTO    017050      4552*  4553*  4555   4602*  4720
TRTVEC= 000014       823*
TSTCNT  001436      1079*  1085*  1088*  1092*  3677
TSTSTR  001444       717   1084*
TST1    002362      1203*
TST10   003204      1509*
TST11   003302      1554*
```

```
TST12   003400          1599#
TST13   003476          1644#
TST14   003574          1689#
TST15   003672          1730#
TST16   003762          1769#
TST17   004052          1820#
TST2    002454          1242#
TST20   004124          1859#
TST21   004206          1896#
TST22   004334          1941#
TST23   004462          1991#
TST24   004550          2013    2016    2019    2037#
TST25   004622          2060    2066#
TST26   004672          2083    2085    2087    2101#
TST27   004726          2127#
TST3    002516          1284#
TST30   005036          2167#
TST31   005246          2191    2227#
TST32   005362          2252    2254    2257    2273#
TST33   005424          2299#
TST34   005472          2331#
TST35   005622          2382#
TST36   005752          2433#
TST37   006102          2484#
TST4    002614          1329#
TST40   006232          2535#
TST41   006362          2586#
TST42   006522          2592    2636#
TST43   006664          2682#
TST44   006724          2713#
TST45   007044          2755#
TST46   007150          2798#
TST47   007224          2823#
TST5    002712          1374#
TST50   007310          2851#
TST51   007402          2880#
TST52   007536          2919#
TST53   007670          2963#
TST54   010162          3047#
TST55   010454          3131#
TST56   010746          3216#
TST57   011240          3300#
TST6    003010          1419#
TST60   011552          3406#
TST61   011744          3469#
TST62   012160          3494    3520    3524    3537#
TST63   012560          3577    3598    3612#
TST7    003106          1464#
TYPBN = 104405          3788    4639#
TYPE  = 104401          1153    1171    1176    3548    3580    3621    3693    3699    3704    3721    3727    3761    3767
                        3773    3779    3783    3808    4002    4035    4066    4074    4110    4127    4129    4132    4134
                        4138    4145    4235    4236    4239    4252    4263    4282    4331    4334    4338    4396    4523
                        4635#
TYPOC = 104402          3698    3709    3726    3766    3772    3778    4118    4142    4238    4636#
```

```
TYPON = 104404          4638#
TYPOS = 104403          4637#
UTEST   001430          1076#  1184*  3691*  3787   4099*
VECTP   001404          1066#  1188*  1189*
VECT1   001402          1065#  1146*  1187*  1188   1190   2727*  3711*  3733*
VECT2   001406          1067#  1190*  1191*  1192   2764*
VECT2P  001410          1068#  1192*  1193*
WSTART= 001460           715   1087#
$APTHD  001000           860    866#
$ASTAT= ****** U        4466   4481
$ATYC   016404          4437   4439#
$ATY1   016360          4435#
$ATY3   016366          4381   4436#
$ATY4   016376          4079   4438#
$AUTOB  001134           897#  1163*  4232   4353
$BASE   001250           961#  1147   3732
$BDADR  001122           892#
$BDDAT  001126           894#  1290*  1291   1305*  1335*  1336   1350*  1380*  1381   1395*  1425*  1426   1440*
                        1470*  1471   1485*  1515*  1516   1530*  1560*  1561   1575*  1605*  1606   1620*  1650*
                        1651   1665*  1695*  1696   1710*  1735*  1737   1752*  1774*  1776   1791*  1829*  1831
                        1834   1876*  1877   1879   1917*  1920   1922   1962*  1965   1967   1999*  2045*  2143*
                        2146   2202*  2205   2207   2352*  2355   2403*  2406   2454*  2457   2505*  2508   2556*
                        2559   2611*  2614   2659*  2662   2808*  2837*  2862*  2865   2901*  2904   2942*  2945
                        2989*  2992   2995   3025*  3028   3029   3073*  3076   3079   3109*  3112   3113   3157*
                        3160   3163   3193*  3196   3197   3242*  3245   3248   3278*  3281   3282   3329*  3332
                        3335   3368*  3371   3372   3420*  3421   3443*  3446   3447   3483*  3484   3506*  3509
                        3523*  3563*  3566   3594*  3597   3636*  3639   4716   4718   4723   4725
$BELL   001164           912#  4066   4102
$BIN    014474          4031*  4034*  4035   4042#
$CDW1   001254           963#
$CHARC  016354          4398*  4408*  4415   4424*  4429#
$CKSWR  015322          4224#  4643
$CMTAG  001100           880#  1096   1097   1105   1111   1112
$CM3  = 000000           910#
$CNTLG  016047          4235   4348#
$CNTLU  016042          4252   4347#
$CPUOP  001222           935#
$CRLF   001171           914#  4074   4102   4110   4129   4134   4138   4263   4347   4397   4432
$DEVCT  001204           926#  1144*  3697   3703*  3708   3717   3725   3734   3736*
$DEVM   001252           962#
$DCAGN  013670          3757   3790   3796#
$ENDAD  013660           846   3792#
$ENDCT  013426          3759#
$ENULL  013674          3799#
$ENV    001214           931#  1157   4076   4376   4444   4468
$ENVM   001215           932#  1133   3673   4378   4383   4446
$EOP    013372          3749#
$EOPCT  013420          3756#  3760
$ERFLG  001103           883#  4061*  4102   4154   4184   4186   4192*  4214
$ERMAX  001115           889#  1113*  4186   4209*  4214
$ERROR  014476          1105   4059#
$ERRPC  001116           890#  4068*  4069*  4070   4102   4116   4716   4718   4720   4722   4723   4725   4727
$ERRTB  001256           980#  4124
$ERRTY  014704          4073   4109#
```

```
$ERTTL  001112          887#  4067*  4102
$ESCAP  001162          911#  1112*  4090  4092  4102  4208*
$ETABL  001214          930#
$ETEND  001256          872    964#
$FATAL  001176          923#  4472*
$FFLG   016624         4435*  4438*  4466  4475*  4483#
$FILLC  001156          908#  4401   4432
$FILLS  001155          907#  4432
$GDADR  001120          891#
$GDDAT  001124          893#  1289*  1291  1304*  1334*  1336  1349*  1379*  1381  1394*  1424*  1426  1439*
                       1469*  1471   1484*  1514*  1516  1529*  1559*  1561  1574*  1604*  1606  1619*  1649*
                       1651   1664*  1694*  1696  1709*  1733*  1734  1737  1751*  1772*  1773  1776  1790*
                       1831*  1832*  1877*  1878*  1906*  1922  1951*  1967  1994*  2040*  2172*  2206*  2207
                       2219   2220   2667*  2809*  2838*  2864*  2885*  2979*  2993  2995  3029  3063*  3077
                       3079   3113   3147*  3161  3163  3197  3232*  3246  3248  3282  3319*  3333  3335
                       3372   3419*  3421  3447  3482*  3484  3496*  3525*  3553*  3626*  4716  4718  4723
$GET42  013650         3789#
$GTSWR  015372         4236#  4641
$HD   = 000001          675    676
$HIBTS  001000          867#
$ICNT   001104          884#  4199*  4200  4202*  4213
$ILLUP  016766         4493   4509   4526#
$INTAG  001135          898#  4264   4353
$ITEMB  001114          888#  4070*  4078  4102  4113
$LF     001172          915#  4102   4338  4347  4432
$LFLG   016623         4476*  4482#
$LPADR  001106          885#  1114*  1205*  4190*  4206*  4211  4213
$LPERR  001110          886#  1115*  1207*  2173*  4089  4190  4207*  4213
$MADR1  001226          948#
$MADR2  001232          952#
$MADR3  001236          955#
$MADR4  001242          958#
$MAIL   001174          868    872    921#  1132  1157  4076  4205  4376
$MAMS1  001224          942#
$MAMS2  001230          950#
$MAMS3  001234          953#
$MAMS4  001240          956#
$MBADR  001002          868#
$MFLG   016622         4436*  4442   4477*  4481#
$MNEW   016065         4239   4351#
$MSGAD  001210          928#  4452*  4455
$MSGLG  001212          929#  4457*
$MSGTY  001174          922#  4450   4458*  4470  4474*
$MSWR   016054         4236   4349#
$MTYP1  001225          943#
$MTYP2  001231          951#
$MTYP3  001235          954#
$MTYP4  001241          957#
$MXCNT  015320         4203   4213#
$NULL   001154          906#  4403   4432
$NWTST= 000001         1200#  1239#  1276#  1278  1321#  1323  1366#  1368  1411#  1413  1456#  1458  1501#
                       1503   1546#  1548  1591#  1593  1636#  1638  1681#  1683  1727#  1766#  1811#  1813
                       1850#  1852   1893#  1938#  1984#  1986  2029#  2031  2063#  2098#  2124#  2158#  2160
                       2224#  2270#  2296#  2323#  2325  2374#  2376  2425#  2427  2476#  2478  2527#  2529
```

```
                    2578#  2580   2633#  2679#  2710#  2752#  2795#  2820#  2848#  2877#  2916#  2960#  3044#
                    3128#  3213#  3297#  3388#  3390   3461#  3463   3534#  3609#
$OCNT   014416      3974#  4003#  4016#
$OMODE  014420      3969#  3973#  3978   3981#  3992#  4018#
$OVER   015304      4167   4183   4191   4201   4210#
$PASS   001202       925#  1132#  1182#  3689   3753#  3754#  3765   3799   4197   4214
$PASTM  001006       870#
$POWER  016774      4524   4529#
$PWRDN  016626      1109   4493#  4521
$PWRMG  016762      4524#
$PWRUP  016700      4503   4509#
$QUES   001170       913#  4102   4282   4331   4347   4432
$RDCHR  015604      4295#  4644
$RDDEC= ****** U    4646
$RDLIN  015724      4323#  4645
$RDOCT= ****** U    4646
$RDSZ = 000010      4316#
$RTNAD  013672      3798#
$R2A  = ****** U    4646
$SAVRE= ****** U    4646
$SAVR6  016772      4502#  4510   4511#  4512#  4528#
$SCOPE  015040      1103   4163#
$SETUP= 000117      1084#  1102   1103   1105   1107   1109   1111   1112   1114   1153   1154   3751   4060
                    4086   4094   4164   4219   4353
$STUP = 177777      1084#
$SVLAD  015250      4175   4204#
$SVPC = 000244       844#   849
$SWR  = 167400       657#   675    681    682    683    684    685    686    687    837#   910    911    912
                    1111   1112   1114   1115   1204   1243   1285   1330   1375   1420   1465   1510   1555
                    1600   1645   1690   1731   1770   1821   1860   1897   1942   1992   2038   2067   2102
                    2128   2168   2228   2274   2300   2332   2383   2434   2485   2536   2587   2637   2683
                    2714   2756   2799   2824   2852   2881   2920   2964   3048   3132   3217   3301   3407
                    3470   3538   3613   3746   3752   3791   3797   3799   4051   4052   4053   4054   4055
                    4064   4071   4083   4087   4102   4155   4156   4157   4158   4159   4166   4178   4180
                    4181   4184   4185   4186   4193   4194   4195   4207   4210   4213   4525
$SWREG  001216       933#  1135
$SWRMK= 000000       687    688    4159   4160   4182
$TESTN  001200       924#  4205*
$TIMES  001160       910#  1111*  1204*  1285*  1330*  1375*  1420*  1465*  1510*  1555*  1600*  1645*  1690*
                    1821*  1860*  1897*  1942*  1992*  2038*  2168*  2332*  2383*  2434*  2485*  2536*  2587*
                    2637*  2714*  2824*  2881*  2964*  3048*  3132*  3217*  3301*  3407*  3470*  3539*  3613*
                    3752*  4193*  4200   4203*  4213
$TKB    001146       903#  3807   3815   4217   4228   4245   4299   4305
$TKS    001144       902#  3813   4217   4226   4242   4266*  4297   4303
$TMP0   001420      1072#  1221*  1255*  2993*  3077*  3161*  3246*  3333*  4723   4725   4727
$TMP1   001422      1073#
$TMP3   001424      1074#  1909*  1910*  1911*  1912   1954*  1955*  1956*  1957   2135*  2136*  2137*  2138
                    2194*  2195*  2196*  2197   2344*  2345*  2346*  2347   2395*  2396*  2397*  2398   2446*
                    2447*  2448*  2449   2497*  2498*  2499*  2500   2548*  2549*  2550*  2551   2603*  2604*
                    2605*  2606   2651*  2652*  2653*  2654   2893*  2894*  2895*  2896   2934*  2935*  2936*
                    2937   2981*  2982*  2983*  2984   3017*  3018*  3019*  3020   3065*  3066*  3067*  3068
                    3101*  3102*  3103*  3104   3149*  3150*  3151*  3152   3185*  3186*  3187*  3188   3234*
                    3235*  3236*  3237   3270*  3271*  3272*  3273   3321*  3322*  3323*  3324   3360*  3361*
                    3362*  3363   3435*  3436*  3437*  3438   3498*  3499*  3500*  3501   3555*  3556*  3557*
```

```
                        3558    3586*   3587*   3588*   3589    3628*   3629*   3630*   3631
$TN    = 000064          675*     838*   1200    1204*   1239    1243*   1276    1285*   1321    1330*   1366    1375*   1411
                        1420*   1456    1465*   1501    1510*   1546    1555*   1591    1600*   1636    1645*   1681    1690*
                        1727    1731*   1766    1770*   1811    1821*   1850    1860*   1893    1897*   1938    1942*   1984
                        1992*   2013    2016    2019    2029    2038*   2060    2063    2067*   2083    2085    2087    2098
                        2102*   2124    2128*   2158    2168*   2191    2224    2228*   2252    2254    2257    2270    2274*
                        2296    2300*   2323    2332*   2374    2383*   2425    2434*   2476    2485*   2527    2536*   2578
                        2587*   2592    2633    2637*   2679    2683*   2710    2714*   2752    2756*   2795    2799*   2820
                        2824*   2848    2852*   2877    2881*   2916    2920*   2960    2964*   3044    3048*   3128    3132*
                        3213    3217*   3297    3301*   3388    3407*   3461    3470*   3494    3520    3524    3534    3538*
                        3577    3598    3609    3613*
$TPB    001152           905*   4421*   4432
$TPFLG  001157           909*   4370    4432
$TPS    001150           904*   4419    4432
$TRAP   017054          1107    4612*
$TRAP2  017076          4623*   4634
$TRP   = 000012         4627*   4636*   4637*   4638*   4639*   4640*   4641    4642*   4643    4644*   4645*   4646*
$TRPAD  017110          4617    4634*
$TSTM   001004           869*
$TSTNM  001102           882*   1206*   3751*   4063    4102    4154    4182    4204*   4205    4210    4214
$TTYIN  016032          4324    4325    4342    4346*
$TYPBN  014422          4028*   4639
$TYPDS= ****** U        4639
$TYPE   016076          4370*   4463    4627    4635
$TYPEC  016310          4268    4400    4407    4414    4419*   4420
$TYPEX  016356          4425    4427    4430*
$TYPOC  014220          3972*   4636
$TYPON  014234          3971    3974*   4638
$TYPOS  014174          3967*   4637
$UNIT   001206           927*
$UNITM  001010           871*
$USWR   001220           934*
$VECT1  001244           959*   1146    3733
$VECT2  001246           960*
$XTSTR  015054          4169*
$$GET4= 000000          3791*
$OFILL  014417          3968*   3972*   3982    4017*
$40CAT= ****** U        4073    4166
.      = 017770          691*    697*    699*    702*    707*    714*    716*    844     845*    847*    849*    850*    856
                         857*    859*    861*    879*    916    1087    1091    1100    1114    1115    1167*   1174*   3551*
                        3583*   3624*   3696*   3724*   3770*   3782*   3799    3800*   3811*   4102    4148*   4213    4214
                        4217    4346*   4347    4353    4432    4484*   4505    4527    4714*
.$ASTA= ****** U        4436    4439
.$X    = 001000          856*    861
```

```
ADTST    1198#  1238
BUFLO    1725#  1726  1765
COMMEN      1#   832#
COUNTM   2320#  2322  2373  2424  2475  2526  2577
CSRDTA   1273#  1274  1319  1364  1409  1454  1499  1544  1589  1634  1679
DFC       982#   988   996  1004  1012  1020  1028  1036  1044  1052  1060
DIVCH    2958#  2959  3043  3127  3212  3296
ECB       660#  1223  1233  1257  1267  1293  1298  1308  1314  1338  1343  1353  1359  1383  1388
         1398   1404  1428  1433  1443  1449  1473  1478  1488  1494  1518  1523  1533  1539  1563
         1568   1578  1584  1608  1613  1623  1629  1653  1658  1668  1674  1698  1703  1713  1719
         1740   1745  1755  1760  1779  1784  1794  1799  1838  1845  1882  1888  1926  1933  1971
         1978   2003  2010  2020  2025  2049  2055  2074  2080  2088  2093  2108  2115  2148  2153
         2183   2188  2209  2213  2247  2260  2265  2288  2309  2314  2361  2366  2412  2417  2463
         2468   2514  2519  2565  2570  2620  2625  2668  2673  2694  2699  2740  2744  2779  2783
         2810   2815  2839  2844  2867  2872  2906  2911  2948  2953  2998  3004  3031  3038  3082
         3088   3115  3122  3166  3172  3199  3206  3251  3257  3284  3291  3338  3344  3374  3381
         3423   3428  3449  3455  3486  3491  3511  3517  3526  3530  3568  3574  3599  3605  3642
         3647   3655  3660  4565  4598
ENDCOM      1#   832#
ENDPAS   3740#  3761
ERROR     726#  1226  1260  1296  1311  1341  1356  1386  1401  1431  1446  1476  1491  1521  1536
         1566   1581  1611  1626  1656  1671  1701  1716  1743  1758  1782  1797  1841  1885  1929
         1974   2007  2023  2052  2077  2091  2111  2151  2186  2212  2250  2263  2291  2312  2364
         2415   2466  2517  2568  2623  2671  2697  2743  2782  2813  2842  2870  2909  2951  3001
         3034   3085  3118  3169  3202  3254  3287  3341  3377  3426  3452  3489  3514  3529  3571
         3602   3645  3658  3856  3897  3931  3935  4568
ESCAPE      1#   832#
GETPRI      1#   832#
GETSWR      1#   832#  1154#
INSTR2   4044#  4094
LOCKM    2709#  2716  2728  2735  2757  2767  2774
MULT        1#   832#
NEWTST      1#   832#  1200  1239  1276  1321  1366  1411  1456  1501  1546  1591  1636  1681  1727
         1766   1811  1850  1893  1938  1984  2029  2063  2098  2124  2158  2224  2270  2296  2323
         2374   2425  2476  2527  2578  2633  2679  2710  2752  2795  2820  2848  2877  2916  2960
         3044   3128  3213  3297  3388  3461  3534  3609
POP         1#   832#  4478  4479  4514  4515
POPSP2    661#  1219  1253  2748  2787  3714  4564
PR        662#  1138
PUSH        1#   832#  4439  4441  4462  4495  4501
RDCLK     663#  1908  1953  2134  2193  2343  2394  2445  2496  2547  2602  2650  2892  2933  2980
         3016   3064  3100  3148  3184  3233  3269  3320  3359  3433  3497  3554  3585  3627
            1#   832#
          727#  1242  1284  1329  1374  1419  1464  1509  1554  1599  1644  1689  1730  1769  1820
         1859   1896  1941  1991  2037  2066  2101  2127  2167  2227  2273  2299  2331  2382  2433
         2484   2535  2586  2636  2682  2713  2755  2798  2823  2851  2880  2919  2963  3047  3131
         3216   3300  3406  3469  3537  3612  3670
SETPRI      1#   832#
SETTRA   4627#  4636  4637  4638  4639  4641  4643  4644  4645
SETUP       1#   832#  1094
SKIP        1#   832#  2013  2016  2019  2060  2083  2085  2087  2191  2252  2254  2257  2592  3494
         3520   3524  3577  3598
SLASH       1#   832#
SPACE     832#
```

```
STARS       1#    832#    842    853    855    862    875    916    919   1200   1202   1239   1241   1276   1283
         1321    1328   1366   1373   1411   1418   1456   1463   1501   1508   1546   1553   1591   1598   1636
         1643    1681   1688   1727   1729   1766   1768   1811   1819   1850   1858   1893   1895   1938   1940
         1984    1990   2029   2036   2063   2065   2098   2100   2124   2126   2158   2166   2224   2226   2270
         2272    2296   2298   2323   2330   2374   2381   2425   2432   2476   2483   2527   2534   2578   2585
         2633    2635   2679   2681   2710   2712   2752   2754   2795   2797   2820   2822   2848   2850   2877
         2879    2916   2918   2960   2962   3044   3046   3128   3130   3213   3215   3297   3299   3388   3405
         3461    3468   3534   3536   3609   3611   3744   3944   4021   4047   4104   4151   4216   4219   4287
         4316    4355   4434   4491   4507   4606
SWRSU       1#     832#  1116#
TRMTRP   4627#
TYPBIN      1#     832#  3787
TYPDEC      1#     832#
TYPNAM      1#     832#  1149
TYPNUM      1#     832#
TYPOCS      1#     654#   832#
TYPOCT      1#     832#  3697   3708   3725   3765   3771   3777   4116   4140   4237
TYPTXT      1#     832#  1171   1175   3548   3580   3621   3693   3699   3704   3721   3727   3761   3767   3773
         3779    3783   3808
ZIOTM    3817#   3821   3862
ZP21     1810#   1813   1852
ZP25     1983#   1986
ZP25A    2028#   2031
ZZ1      1272#   1278   1323   1368   1413   1458   1503   1548   1593   1638   1683   2158#  2160   2322#  2325
         2373#   2376   2424#  2427   2475#  2478   2526#  2529   2577#  2580   3387#  3390   3460#  3463
$$CMRE    873#
$$CMTM    873#
$$ESCA      1#     832#
$$NEWT      1#     832#  1200   1239   1276   1321   1366   1411   1456   1501   1546   1591   1636   1681   1727
         1766    1811   1850   1893   1938   1984   2029   2063   2098   2124   2158   2224   2270   2296   2323
         2374    2425   2476   2527   2578   2633   2679   2710   2752   2795   2820   2848   2877   2916   2960
         3044    3128   3213   3297   3388   3461   3534   3609
$$SET    4627#   4636   4637   4638   4639   4641   4643   4644   4645
$$SETM   1132#
$$SKIP      1#     832#  2013   2016   2019   2060   2083   2085   2087   2191   2252   2254   2257   2592   3494
         3520    3524   3577   3598
.EQUAT      1#     654#   722
.HEADE      1#     653#   665
.KT11       1#
.SETTR   653#
.SETUP      1#     653#  1084
.SWRHI      1#     654#   677
.SWRLO   688#
.TRMTR   653#
.$ACTI      1#     656#   840
.$APTB      1#     917#
.$APTH      1#     656#   851
.$APTY      1#     656#  4432
.$ASTA      1#
.$CATC      1#     654#
.$CMTA      1#     654#   873
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
```

```
.$EOP       1#    655#   3742
.$ERRO      1#    655#   4045
.$ERRT      1#    655#   4102
.$MULT      1#
.$POWE      1#    654#   4489
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    653#
.$READ      1#    655#   4214
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    655#   4149
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    653#   4604
.$TYPB      1#    653#   4019
.$TYPD      1#    655#
.$TYPE      1#    655#   4353
.$TYPO      1#    654#   3942
.$40CA      1#
.1170       1#
```

```
ADCB    4034
ADD     1189  1191  1193  1197  1220  1254  2749  2788  3682  3711  3715  3970  3980  4124  4253
        4262  4389  4449  4461  4473  4564
ASL     3679  4121  4122  4123  4276  4277  4278  4616
ASR     4456
BEQ     1134  1158  1170  1292  1306  1337  1351  1382  1396  1427  1441  1472  1486  1517  1531
        1562  1576  1607  1621  1652  1666  1697  1711  1738  1753  1777  1792  1836  1880  1924
        1969  2001  2016  2019  2047  2060  2085  2180  2208  2230  2254  2286  2592  2866  2905
        2997  3030  3081  3114  3165  3198  3250  3283  3337  3373  3422  3448  3485  3510  3524
        3616  3641  3678  3718  3790  3855  3934  3997  4033  4062  4065  4088  4091  4126  4131
        4144  4181  4183  4185  4189  4198  4233  4260  4275  4379  4392  4427  4443  4447  4467
        4469
BGE     4201
BGT     3757  4004  4272  4313
BHI     4187
BIC     1187  1303  1348  1393  1438  1483  1528  1573  1618  1663  1708  1750  1789  1910  1955
        2136  2195  2345  2396  2447  2498  2549  2604  2652  2857  2894  2935  2982  3018  3066
        3102  3150  3186  3235  3271  3322  3361  3436  3499  3556  3587  3629  3754  3994  4099
        4229  4246  4273  4300  4306  4314
BIS     1288  1333  1378  1423  1468  1513  1558  1603  1648  1693  1904  1911  1916  1949  1956
        1961  2017  2061  2069  2104  2131  2132  2137  2142  2175  2177  2196  2201  2234  2240
        2242  2279  2281  2303  2305  2346  2351  2397  2402  2448  2453  2499  2504  2550  2555
        2605  2610  2642  2644  2653  2658  2686  2726  2766  2802  2803  2828  2854  2855  2859
        2888  2895  2900  2923  2924  2927  2936  2941  2968  2969  2974  2983  2988  3011  3019
        3024  3052  3053  3058  3067  3072  3095  3103  3108  3136  3137  3142  3151  3156  3179
        3187  3192  3221  3222  3227  3236  3241  3264  3272  3277  3305  3306  3312  3323  3328
        3352  3362  3367  3414  3418  3437  3442  3477  3481  3500  3505  3557  3562  3588  3593
        3630  3635  3691  3851  3853  3892  3928  3999  4000  4280
BISB    4113
BIT     2018  2086  2105  2181  2285  2307  2589  2805  2834  2865  3933  4064  4071  4087  4166
        4180  4188  4195
BITB    1133  4378  4383  4415  4446
BLE     4556
BLOS    4326
BLT     4005  4270  4311  4406
BMI     2072  2245  2257  2359  2410  2461  2512  2563  2618  2692  3654  3674  3896  3930
BNE     1100  1123  1152  1156  1160  2087  2106  2147  2182  2221  2259  2308  2341  2356  2392
        2407  2443  2458  2494  2509  2545  2560  2600  2615  2648  2663  2689  2806  2831  2835
        2891  2931  2946  2976  3013  3060  3097  3144  3181  3229  3266  3314  3317  3354  3356
        3416  3479  3541  3567  3598  3690  3995  4072  4077  4096  4114  4136  4167  4196  4225
        4231  4251  4258  4265  4302  4308  4330  4336  4377  4384  4386  4394  4402  4416  4423
        4445  4451  4454  4471  4513
BPL     2665  3814  3993  4084  4227  4243  4298  4304  4371  4420
BR      1086  1090  1125  1162  1165  1172  1177  1218  1252  1301  1346  1391  1436  1481  1526
        1571  1616  1661  1706  1748  1787  2013  2083  2191  2216  2252  2747  2786  3007  3091
        3175  3260  3347  3431  3494  3520  3549  3577  3581  3622  3650  3694  3700  3705  3722
        3728  3762  3768  3774  3780  3784  3809  3857  3898  3936  3971  3986  4007  4037  4082
        4119  4146  4169  4175  4178  4191  4194  4254  4281  4283  4309  4332  4373  4399  4409
        4418  4425  4437  4459  4505  4527  4561
CLC     4036
CLR     1089  1093  1098  1111  1112  1132  1144  1180  1181  1182  1287  1304  1332  1349  1377
        1394  1422  1439  1467  1484  1512  1529  1557  1574  1602  1619  1647  1664  1692  1709
        1732  1751  1771  1790  1823  1862  1899  1944  1994  2040  2118  2129  2130  2170  2171
        2218  2232  2237  2275  2301  2318  2335  2336  2338  2370  2386  2387  2389  2421  2437
```

	2438	2440	2472	2488	2489	2491	2523	2539	2540	2542	2574	2594	2595	2597	2629
	2639	2640	2645	2667	2676	2684	2687	2722	2750	2763	2789	2800	2826	2829	2875
	2883	2884	2885	2889	2914	2921	2922	2956	2966	2967	3041	3050	3051	3125	3134
	3135	3209	3219	3220	3294	3303	3304	3384	3409	3410	3472	3473	3496	3522	3525
	3545	3618	3626	3738	3751	3752	3848	3849	3889	3984	4112	4193	4208	4240	4241
	4511														
CLRB	1832	1878	4192	4337	4398	4424	4475	4476	4477						
CMP	1099	1122	1159	1291	1336	1381	1426	1471	1516	1561	1606	1651	1696	1737	1776
	1922	1967	2207	2995	3029	3079	3113	3163	3197	3248	3282	3335	3372	3421	3447
	3484	3677	3717	3854	4176	4200	4224	4230	4250	4257	4269	4271	4301	4307	4310
	4312	4325	4555												
CMPB	1157	4076	4182	4186	4232	4264	4329	4335	4376	4391	4393	4401	4422	4426	4444
DEC	1874	2258	3012	3096	3180	3265	3316	3353	3355	3755	4097	4120			
DECB	3992	4003	4405	4408											
EMT	726														
HALT	4085	4372	4504	4526	4558										
INC	1084	1151	1864	2206	2340	2391	2442	2493	2544	2599	2647	2801	2930	2975	3059
	3143	3228	3313	3415	3478	3680	3703	3735	3753	3998	4006	4067	4095	4199	4279
	4474	4512													
INCB	2688	2830	2890	4061	4204	4428									
IOT	727														
JMP	708	709	710	711	712	715	717	3542	3712	3797					
JSR	3552	3584	3625	3792	4073	4079	4268	4381	4400	4407	4414	4463			
MOV	1085	1088	1092	1097	1101	1103	1104	1105	1106	1107	1108	1109	1110	1114	1115
	1118	1119	1120	1121	1126	1128	1129	1130	1135	1139	1140	1145	1146	1147	1183
	1184	1188	1190	1192	1196	1204	1205	1207	1211	1212	1221	1236	1245	1246	1255
	1270	1285	1289	1290	1305	1330	1334	1335	1350	1375	1379	1380	1395	1420	1424
	1425	1440	1465	1469	1470	1485	1510	1514	1515	1530	1555	1559	1560	1575	1600
	1604	1605	1620	1645	1649	1650	1665	1690	1694	1695	1710	1733	1734	1735	1752
	1772	1773	1774	1791	1821	1829	1831	1860	1876	1877	1897	1900	1906	1908	1909
	1912	1917	1919	1942	1945	1951	1953	1954	1957	1962	1964	1992	1995	1999	2038
	2041	2045	2068	2103	2134	2135	2138	2143	2145	2168	2172	2173	2193	2194	2197
	2202	2204	2219	2235	2238	2277	2302	2332	2337	2343	2344	2347	2352	2354	2383
	2388	2394	2395	2398	2403	2405	2434	2439	2445	2446	2449	2454	2456	2485	2490
	2496	2497	2500	2505	2507	2536	2541	2547	2549	2551	2556	2558	2587	2596	2602
	2603	2606	2611	2613	2637	2650	2651	2654	2659	2661	2685	2714	2717	2718	2723
	2725	2727	2729	2730	2736	2737	2758	2759	2764	2765	2768	2769	2775	2776	2808
	2809	2824	2827	2837	2838	2853	2862	2864	2881	2887	2892	2893	2896	2901	2903
	2925	2933	2934	2937	2942	2944	2964	2972	2979	2980	2981	2984	2989	2991	2993
	3008	3016	3017	3020	3025	3027	3048	3056	3063	3064	3065	3068	3073	3075	3077
	3092	3100	3101	3104	3109	3111	3132	3140	3147	3148	3149	3152	3157	3159	3161
	3176	3184	3185	3188	3193	3195	3217	3225	3232	3233	3234	3237	3242	3244	3246
	3261	3269	3270	3273	3278	3280	3301	3308	3310	3319	3320	3321	3324	3329	3331
	3333	3348	3349	3359	3360	3363	3368	3370	3407	3411	3413	3419	3420	3434	3435
	3438	3443	3445	3470	3474	3476	3482	3483	3497	3498	3501	3506	3508	3523	3539
	3546	3547	3553	3554	3555	3558	3563	3565	3585	3586	3589	3594	3596	3613	3619
	3620	3627	3628	3631	3636	3638	3683	3684	3697	3708	3710	3716	3725	3732	3733
	3734	3736	3739	3758	3765	3771	3777	3787	3789	3850	3852	3890	3891	3927	3967
	3975	3976	3977	3983	3990	4008	4009	4010	4011	4012	4028	4029	4038	4039	4040
	4063	4068	4089	4092	4111	4116	4125	4130	4135	4137	4141	4171	4172	4174	4177
	4190	4202	4203	4206	4207	4210	4211	4237	4261	4266	4295	4296	4323	4324	4339
	4340	4341	4342	4374	4375	4380	4388	4403	4440	4441	4448	4452	4457	4458	4460
	4462	4472	4478	4479	4493	4494	4495	4496	4497	4498	4499	4500	4501	4502	4503
	4509	4510	4514	4515	4516	4517	4518	4519	4520	4521	4522	4552	4563	4612	4613

```
            4617  4623  4624
MOVB        1113  1163  1206  1824  1868  3968  3969  3972  3973  3974  3978  3981  3982  4001  4031
            4070  4078  4205  4209  4228  4245  4299  4305  4328  4333  4385  4413  4421  4435  4436
            4438  4615
NEG         3979
NOP         1203  2734  2773  3750  3793  3794  3795  3893  3894
RESV        1997  2043  3791
ROL         3985  3987  3988  3989  3991  4032
RTI         1127  1141  2719  2731  2738  2760  2770  2777  4013  4041  4100  4212  4267  4315  4343
            4390  4525  4601  4625
RTS         3816  4139  4430  4480  4618
SEC         4030
SUB         4069  4455  4553
TRAP        4627  4636  4637  4638  4639  4641  4643  4644  4645
TST         1155  1169  1214  1248  1920  1965  2015  2059  2071  2084  2146  2179  2205  2220  2229
            2233  2253  2355  2406  2457  2508  2559  2614  2662  2904  2945  2992  3028  3076  3112
            3160  3196  3245  3281  3332  3371  3446  3509  3540  3566  3597  3615  3639  3653  3686
            3689  3895  3929  3996  4083  4090  4143  4173  4197  4259  4274  4387  4395  4417  4450
            4468  4470  4614
TSTB        1834  1879  2244  2255  2357  2408  2459  2510  2561  2616  2664  2691  3673  3807  3813
            3815  4184  4226  4242  4297  4303  4370  4419  4442  4453  4466
.ASCII       913   914
.ASCIZ       912   915  1167  1174  1179  3551  3583  3624  3696  3702  3707  3724  3730  3764  3770
            3776  3782  3786  3811  4147  4347  4348  4349  4351  4529  4646  4651  4655  4659  4662
            4666  4669  4673  4678  4682  4687  4691  4695  4700  4703  4709
.BLKB       4346
.BYTE        882   883   888   889   897   898   906   907   908   909   931   932   942   943   950
             95i   953   954   956   957  3799  4014  4015  4016  4017  4042  4080  4081  4344  4345
            4481  4482  4483
.DSABL      4284
.ENABL         1   651   652  4217
.END        4731
.ENDC        670   684   686   687   688   726   818   832   843   847   849   854   856   863   876
             880   882   910   911   912   913   917   920   942   950   953   956   959   960   961
             962   963   966  1084  1101  1102  1105  1107  1109  1111  1112  1114  1116  1137  1153
            1159  1165  1167  1174  1179  1201  1202  1203  1204  1205  1206  1209  1240  1241  1242
            1243  1277  1278  1283  1284  1285  1286  1322  1323  1328  1329  1330  1331  1367  1368
            1373  1374  1375  1376  1412  1413  1418  1419  1420  1421  1457  1458  1463  1464  1465
            1466  1502  1503  1508  1509  1510  1511  1547  1548  1553  1554  1555  1556  1592  1593
            1598  1599  1600  1601  1637  1638  1643  1644  1645  1646  1682  1683  1688  1689  1690
            1691  1728  1729  1730  1731  1767  1768  1769  1770  1812  1813  1819  1820  1821  1822
            1851  1852  1858  1859  1860  1861  1894  1895  1896  1897  1898  1939  1940  1941  1942
            1943  1985  1986  1990  1991  1992  1993  2014  2017  2020  2030  2031  2036  2037  2038
            2039  2061  2064  2065  2066  2067  2084  2086  2088  2099  2100  2101  2102  2125  2126
            2127  2128  2159  2160  2166  2167  2168  2169  2192  2225  2226  2227  2228  2253  2255
            2258  2271  2272  2273  2274  2297  2298  2299  2300  2324  2325  2330  2331  2332  2333
            2334  2343  2375  2376  2381  2382  2383  2384  2385  2394  2426  2427  2432  2433  2434
            2435  2436  2445  2477  2478  2483  2484  2485  2486  2487  2496  2528  2529  2534  2535
            2536  2537  2538  2547  2579  2580  2585  2586  2587  2588  2593  2602  2634  2635  2636
            2637  2638  2680  2681  2682  2683  2711  2712  2713  2714  2715  2753  2754  2755  2756
            2796  2797  2798  2799  2821  2822  2823  2824  2825  2849  2850  2851  2852  2878  2879
            2880  2881  2882  2917  2918  2919  2920  2961  2962  2963  2964  2965  2971  2978  2998
            3008  3014  3045  3046  3047  3048  3049  3055  3062  3082  3092  3098  3129  3130  3131
            3132  3133  3139  3146  3166  3176  3182  3214  3215  3216  3217  3218  3224  3231  3251
```

```
            3261  3267  3298  3299  3300  3301  3302  3309  3318  3338  3349  3357  3389  3390  3405
            3406  3407  3408  3462  3463  3468  3469  3470  3471  3495  3521  3525  3535  3536  3537
            3538  3551  3578  3583  3599  3610  3611  3612  3613  3614  3624  3696  3702  3707  3724
            3730  3745  3746  3748  3751  3757  3760  3761  3764  3770  3776  3782  3786  3789  3791
            3797  3799  3800  3811  3945  4022  4048  4051  4061  4068  4073  4074  4075  4083  4094
            4102  4105  4120  4149  4152  4155  4160  4166  4168  4179  4182  4183  4184  4186  4188
            4195  4199  4204  4206  4210  4213  4214  4217  4218  4220  4248  4284  4288  4316  4317
            4324  4326  4329  4331  4347  4353  4356  4385  4435  4436  4439  4466  4481  4492  4501
            4502  4508  4514  4515  4525  4532  4607  4613  4616  4635  4636  4637  4638  4639  4640
            4641  4642  4643  4644  4645  4646
.EQUIV       726   727   735   780   781   782   783   784   785   786   787   788   789   808   809
             810   811   812   813   814   815   816   817
.EVEN        920  1167  1174  1179  3551  3583  3624  3696  3702  3707  3724  3730  3764  3770  3776
            3782  3786  3800  3811  4148  4484  4531  4714
.IF          666   684   685   686   687   688   724   790   818   842   845   847   853   855   862
             875   879   881   910   911   912   916   917   919   942   950   953   956   959   960
             961   962   963   964   966  1084  1096  1101  1103  1105  1107  1109  1111  1112  1114
            1132  1152  1153  1154  1157  1166  1173  1178  1199  1200  1202  1204  1205  1206  1239
            1241  1243  1276  1278  1283  1285  1286  1321  1323  1328  1330  1331  1366  1368  1373
            1375  1376  1411  1413  1418  1420  1421  1456  1458  1463  1465  1466  1501  1503  1508
            1510  1511  1546  1548  1553  1555  1556  1591  1593  1598  1600  1601  1636  1638  1643
            1645  1646  1681  1683  1688  1690  1691  1727  1729  1731  1766  1768  1770  1811  1813
            1819  1821  1822  1850  1852  1858  1860  1861  1893  1895  1897  1898  1938  1940  1942
            1943  1984  1986  1990  1992  1993  2013  2016  2019  2029  2031  2036  2038  2039  2060
            2063  2065  2067  2083  2085  2087  2098  2100  2102  2124  2126  2128  2158  2160  2166
            2168  2169  2191  2224  2226  2228  2252  2254  2257  2270  2272  2274  2296  2298  2300
            2323  2325  2330  2332  2333  2334  2338  2374  2376  2381  2383  2384  2385  2389  2425
            2427  2432  2434  2435  2436  2440  2476  2478  2483  2485  2486  2487  2491  2527  2529
            2534  2536  2537  2538  2542  2578  2580  2585  2587  2588  2589  2592  2597  2633  2635
            2637  2638  2679  2681  2683  2710  2712  2714  2715  2752  2754  2756  2795  2797  2799
            2820  2822  2824  2825  2848  2850  2852  2877  2879  2881  2882  2916  2918  2920  2960
            2962  2964  2965  2971  2978  2997  3008  3014  3044  3046  3048  3049  3055  3062  3081
            3092  3098  3128  3130  3132  3133  3139  3146  3165  3176  3182  3213  3215  3217  3218
            3224  3231  3250  3261  3267  3297  3299  3301  3302  3308  3316  3337  3348  3355  3388
            3390  3405  3407  3408  3461  3463  3468  3470  3471  3494  3520  3524  3534  3536  3538
            3550  3577  3582  3598  3609  3611  3613  3614  3623  3695  3701  3706  3723  3729  3744
            3745  3746  3747  3748  3750  3756  3759  3761  3763  3769  3775  3781  3785  3789  3791
            3797  3799  3800  3810  3944  4021  4047  4050  4061  4064  4071  4073  4074  4076  4083
            4087  4094  4102  4104  4119  4135  4151  4154  4159  4165  4166  4178  4180  4181  4182
            4184  4185  4186  4195  4197  4205  4207  4212  4213  4214  4216  4218  4219  4220  4248
            4287  4288  4316  4324  4325  4329  4330  4346  4347  4353  4355  4376  4434  4436  4439
            4466  4481  4491  4501  4502  4507  4514  4515  4523  4525  4529  4606  4612  4616  4627
            4636  4637  4638  4639  4641  4643  4644  4645  4646
.IFF          684   686   687   688   724   843   847   849   854   856   863   876   879   882   910
             917   920  1101  1152  1153  1200  1201  1202  1203  1204  1209  1239  1240  1241  1242
            1243  1277  1278  1284  1285  1322  1323  1329  1330  1367  1368  1374  1375  1412  1413
            1419  1420  1457  1458  1464  1465  1502  1503  1509  1510  1547  1548  1554  1555  1592
            1593  1599  1600  1637  1638  1644  1645  1682  1683  1689  1690  1728  1729  1730  1731
            1767  1768  1769  1770  1812  1813  1820  1821  1851  1852  1859  1860  1894  1895  1896
            1897  1939  1940  1941  1942  1985  1986  1991  1992  2014  2017  2020  2030  2031  2037
            2038  2061  2064  2065  2066  2067  2084  2086  2088  2099  2100  2101  2102  2125  2126
            2127  2128  2159  2160  2167  2168  2192  2225  2226  2227  2228  2253  2255  2258  2271
            2272  2273  2274  2297  2298  2299  2300  2324  2325  2331  2332  2338  2375  2376  2382
            2383  2389  2426  2427  2433  2434  2440  2477  2478  2484  2485  2491  2528  2529  2535
```

```
          2536  2542  2579  2580  2586  2587  2593  2597  2634  2635  2636  2637  2680  2681  2682
          2683  2711  2712  2713  2714  2753  2754  2755  2756  2796  2797  2798  2799  2821  2822
          2823  2824  2849  2850  2851  2852  2878  2879  2880  2881  2917  2918  2919  2920  2961
          2962  2963  2964  2997  3045  3046  3047  3048  3081  3129  3130  3131  3132  3165  3214
          3215  3216  3217  3250  3298  3299  3300  3301  3338  3389  3390  3406  3407  3462  3463
          3469  3470  3495  3521  3525  3535  3536  3537  3538  3578  3599  3610  3611  3612  3613
          3745  3747  3750  3757  3760  3799  3945  4022  4048  4050  4064  4094  4105  4120  4149
          4152  4179  4182  4183  4186  4213  4214  4217  4220  4288  4290  4295  4316  4317  4326
          4330  4347  4356  4435  4492  4508  4525  4607  4613
.IFT      1167  1174  1179  3551  3583  3624  3696  3702  3707  3724  3730  3764  3770  3776  3782
          3786  3811  4074  4194  4290  4295
.IFTF     1167  1174  1179  3551  3583  3624  3696  3702  3707  3724  3730  3764  3770  3776  3782
          3786  3811  4073  4192  4235  4288  4291
.IIF       665   670   675   676   681   682   683   684   687   688   916   920  1102  1105  1111
          1112  1114  1115  1153  3698  3709  3726  3746  3751  3752  3766  3772  3778  3788  3799
          3800  4051  4052  4053  4054  4055  4060  4086  4094  4102  4117  4142  4155  4156  4157
          4158  4159  4160  4164  4193  4194  4210  4213  4214  4217  4238  4339  4347  4353  4432
          4635  4636  4637  4638  4639  4641  4643  4644  4645
.IRP      1084  1200  1239  1276  1321  1366  1411  1456  1501  1546  1591  1636  1681  1727  1766
          1811  1850  1893  1938  1984  2029  2063  2098  2124  2158  2224  2270  2296  2323  2374
          2425  2476  2527  2578  2633  2679  2710  2752  2795  2820  2848  2877  2916  2960  3044
          3128  3213  3297  3388  3461  3534  3609  3750  4094  4165  4440  4441  4462  4478  4479
          4495  4501  4514  4515
.LIST        1   650   687   697   832   910   917   920  1084  1116  1153  1154  1167  1174  1179
          1200  1204  1224  1226  1234  1236  1239  1243  1258  1260  1268  1270  1275  1276  1285
          1287  1294  1296  1299  1301  1309  1311  1315  1317  1320  1321  1330  1332  1339  1341
          1344  1346  1354  1356  1360  1362  1365  1366  1375  1377  1384  1386  1389  1391  1399
          1401  1405  1407  1410  1411  1420  1422  1429  1431  1434  1436  1444  1446  1450  1452
          1455  1456  1465  1467  1474  1476  1479  1481  1489  1491  1495  1497  1500  1501  1510
          1512  1519  1521  1524  1526  1534  1536  1540  1542  1545  1546  1555  1557  1564  1566
          1569  1571  1579  1581  1585  1587  1590  1591  1600  1602  1609  1611  1614  1616  1624
          1626  1630  1632  1635  1636  1645  1647  1654  1656  1659  1661  1669  1671  1675  1677
          1680  1681  1690  1692  1699  1701  1704  1706  1714  1716  1720  1722  1727  1731  1741
          1743  1746  1748  1756  1758  1761  1763  1766  1770  1780  1782  1785  1787  1795  1797
          1800  1802  1811  1821  1839  1841  1846  1848  1850  1860  1883  1885  1889  1891  1893
          1897  1927  1929  1934  1936  1938  1942  1972  1974  1979  1981  1984  1992  2004  2006
          2011  2013  2021  2023  2026  2028  2029  2038  2050  2052  2056  2058  2063  2067  2075
          2077  2081  2083  2089  2091  2094  2096  2098  2102  2109  2111  2116  2118  2124  2128
          2149  2151  2154  2156  2158  2168  2184  2186  2189  2191  2210  2212  2214  2216  2224
          2228  2248  2250  2261  2263  2266  2268  2270  2274  2289  2291  2296  2300  2310  2312
          2315  2317  2323  2332  2362  2364  2367  2369  2374  2383  2413  2415  2418  2420  2425
          2434  2464  2466  2469  2471  2476  2485  2515  2517  2520  2522  2527  2536  2566  2568
          2571  2573  2578  2587  2621  2623  2626  2628  2633  2637  2669  2671  2674  2676  2679
          2683  2695  2697  2700  2702  2710  2714  2741  2743  2745  2747  2752  2756  2780  2782
          2784  2786  2795  2799  2811  2813  2816  2818  2820  2824  2840  2842  2845  2847  2848
          2852  2868  2870  2873  2875  2877  2881  2907  2909  2912  2914  2916  2920  2949  2951
          2954  2956  2960  2964  2999  3001  3005  3007  3032  3034  3039  3041  3044  3048  3083
          3085  3089  3091  3116  3118  3123  3125  3128  3132  3167  3169  3173  3175  3200  3202
          3207  3209  3213  3217  3252  3254  3258  3260  3285  3287  3292  3294  3297  3301  3339
          3341  3345  3347  3375  3377  3382  3384  3388  3407  3424  3426  3429  3431  3450  3452
          3456  3458  3461  3470  3487  3489  3492  3494  3512  3514  3518  3520  3527  3529  3531
          3533  3534  3538  3551  3569  3571  3575  3577  3583  3600  3602  3606  3608  3609  3613
          3624  3643  3645  3648  3650  3656  3658  3661  3663  3696  3702  3707  3724  3730  3751
          3764  3770  3776  3782  3786  3791  3811  4094  4159  4316  4566  4568  4599  4601  4627
```

```
                4635  4636  4637  4638  4639  4640  4641  4642  4643  4644  4645  4646
.MACRO             1   660   661   662   663   688   873   982  1132  1198  1272  1273  1725  1810  1983
                2028  2158  2320  2322  2373  2424  2475  2526  2577  2709  2958  3387  3460  3740  3817
                4044  4627
.MCALL           653   654   655   656   832   917  1116  1154
.MEXIT           965
.NLIST             1   649   687   697   832   910   917   920  1084  1116  1153  1154  1167  1174  1179
                1200  1204  1224  1226  1234  1236  1239  1243  1258  1260  1268  1270  1275  1276  1285
                1287  1294  1296  1299  1301  1309  1311  1315  1317  1320  1321  1330  1332  1339  1341
                1344  1346  1354  1356  1360  1362  1365  1366  1375  1377  1384  1386  1389  1391  1399
                1401  1405  1407  1410  1411  1420  1422  1429  1431  1434  1436  1444  1446  1450  1452
                1455  1456  1465  1467  1474  1476  1479  1481  1489  1491  1495  1497  1500  1501  1510
                1512  1519  1521  1524  1526  1534  1536  1540  1542  1545  1546  1555  1557  1564  1566
                1569  1571  1579  1581  1585  1587  1590  1591  1600  1602  1609  1611  1614  1616  1624
                1626  1630  1632  1635  1636  1645  1647  1654  1656  1659  1661  1669  1671  1675  1677
                1680  1681  1690  1692  1699  1701  1704  1706  1714  1716  1720  1722  1727  1731  1741
                1743  1746  1748  1756  1758  1761  1763  1766  1770  1780  1782  1785  1787  1795  1797
                1800  1802  1811  1821  1839  1841  1846  1848  1850  1860  1883  1885  1889  1891  1893
                1897  1927  1929  1934  1936  1938  1942  1972  1974  1979  1981  1984  1992  2004  2006
                2011  2013  2021  2023  2026  2028  2029  2038  2050  2052  2056  2058  2063  2067  2075
                2077  2081  2083  2089  2091  2094  2096  2098  2102  2109  2111  2116  2118  2124  2128
                2149  2151  2154  2156  2158  2168  2184  2186  2189  2191  2210  2212  2214  2216  2224
                2228  2248  2250  2261  2263  2266  2268  2270  2274  2289  2291  2296  2300  2310  2312
                2315  2317  2323  2332  2362  2364  2367  2369  2374  2383  2413  2415  2418  2420  2425
                2434  2464  2466  2469  2471  2476  2485  2515  2517  2520  2522  2527  2536  2566  2568
                2571  2573  2578  2587  2621  2623  2626  2628  2633  2637  2669  2671  2674  2676  2679
                2683  2695  2697  2700  2702  2710  2714  2741  2743  2745  2747  2752  2756  2780  2782
                2784  2786  2795  2799  2811  2813  2816  2818  2820  2824  2840  2842  2845  2847  2848
                2852  2868  2870  2873  2875  2877  2881  2907  2909  2912  2914  2916  2920  2949  2951
                2954  2956  2960  2964  2999  3001  3005  3007  3032  3034  3039  3041  3044  3048  3083
                3085  3089  3091  3116  3118  3123  3125  3128  3132  3167  3169  3173  3175  3200  3202
                3207  3209  3213  3217  3252  3254  3258  3260  3285  3287  3292  3294  3297  3301  3339
                3341  3345  3347  3375  3377  3382  3384  3388  3407  3424  3426  3429  3431  3450  3452
                3456  3458  3461  3470  3487  3489  3492  3494  3512  3514  3518  3520  3527  3529  3531
                3533  3534  3538  3551  3569  3571  3575  3577  3583  3600  3602  3606  3608  3609  3613
                3624  3643  3645  3648  3650  3656  3658  3661  3663  3696  3702  3707  3724  3730  3751
                3764  3770  3776  3782  3786  3791  3811  4094  4159  4316  4566  4568  4599  4601  4627
                4635  4636  4637  4638  4639  4640  4641  4642  4643  4644  4645  4646
.PAGE            873   966  1275  1320  1365  1410  1455  1500  1545  1590  1635  1680
.RADIX          1275  1287  1320  1332  1365  1377  1410  1422  1455  1467  1500  1512  1545  1557  1590
                1602  1635  1647  1680  1692  1724
.REM               1
.REPT            697  2795
.SBTTL           677   689   722   840   851   873   917   966  1095  1149  1154  1200  1239  1276  1321
                1366  1411  1456  1501  1546  1591  1636  1681  1727  1766  1806  1807  1808  1811  1850
                1893  1938  1984  2029  2063  2098  2120  2121  2122  2124  2158  2224  2270  2296  2323
                2374  2425  2476  2527  2578  2633  2679  2705  2706  2707  2710  2752  2791  2792  2793
                2795  2820  2848  2877  2916  2960  3044  3128  3213  3297  3388  3461  3534  3609  3742
                3819  3860  3901  3938  3939  3940  3942  4019  4045  4102  4149  4214  4353  4432  4489
                4604  4627
.TITLE           665
.WORD            697   698   700   701   703   848   867   868   869   870   871   872   881   884   885
                886   887   890   891   892   893   894   895   896   899   900   901   922   923   924
                925   926   927   928   929   933   934   935   948   952   955   958   959   960   961
```

```
  962   963  1053  1064  1065  1066  1067  1068  1069  1070  1071  1072  1073  1074  1075
 1076  1077  1078  1079  1080  1081  3756  3759  3798  4018  4128  4133  4382  4429  4464
 4524  4602  4603  4634  4716  4718  4720  4722  4723  4725  4727  4729
```

ERRORS DETECTED: 0

```
*,DVKWAB.SEQ/CRF/SOL/NL:TOC=DSKZ:DVKWAB.SML,DVKWAB.P11
RUN-TIME:  20 24 2 SECONDS
CORE USED:  34K
```

Spooler runtime 4 Seconds, 40 KCS, 460 disk reads, 3 disk writes, 114 pages