PDP11

UNIBUS EXERCISER MODULE

CZKUBB0

AH-8860B-MC

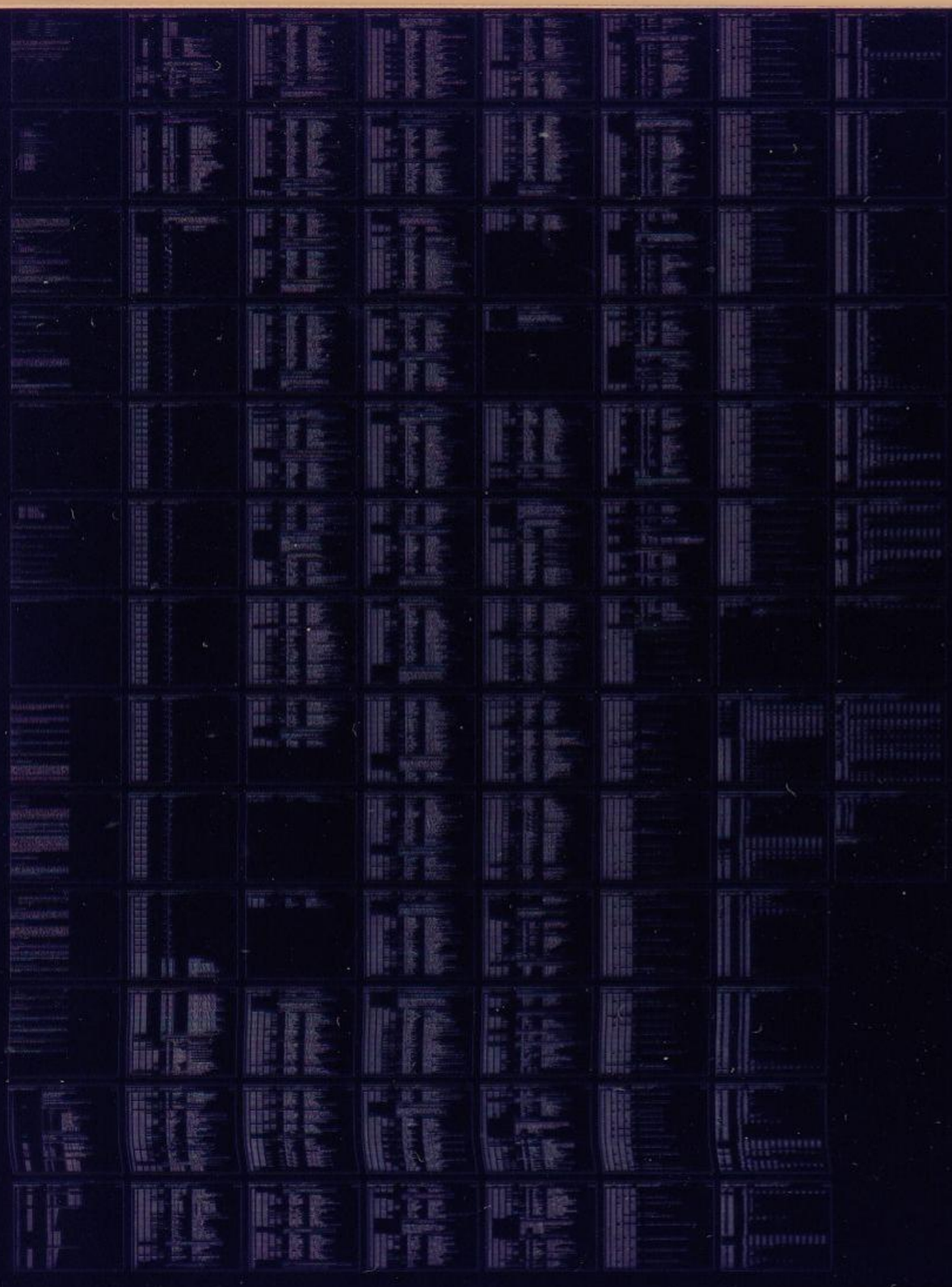
COPYRIGHT © 75-78

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: AC-8859B-MC

PRODUCT NAME: CZKUBB0 UNIBUS EXERCISER MODULE DIAGNOSTIC

DATE CREATED: 1-APRIL-78

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: WARREN SALTZ

MODIFIED BY: BILL SCHLITZKUS

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Copyright (C) 1975,1978 by Digital Equipment Corporation

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

Table of Contents

1.0 Abstract

2.0 Requirements
2.1 Equipment
2.2 Preliminary Requirements
2.3 Execution time

3.0 Starting Address

4.0 Program Control and Operator Action

5.0 Switch Options

6.0 Program Description

7.0 Error Reporting

8.0 Handlers and Common Routines
8.1 Trap Handler
8.2 Scope Handler
8.3 Error Handler
8.4 Trap Catcher
8.5 Power Down and Up Routines
8.6 CLRREG Routine
8.7 RCATCH Routine
8.8 CRDY Routine
8.9 DINT Routine
8.10 RVEC Routine
8.11 TERRPC ROUTINE

1.0 Abstract

The Unibus Exercisor (UBE) module diagnostic is comprised of a series of tests that check all programmatically accessable areas of the exercisors (95%). The tests are arranged in a logical order such that simpler functions are examined first followed by the more complex ones. The tests build on one another such that the present test will use hardware previously tested. This should provide a very effective degree of fault isolation.

The program is written to test a maximum of four UBE's at one time and is intended to run in a stand-alone environment.

2.0 Requirements

2.1 Equipment

1. A working PDP-11 and Unibus
2. A working Teletype
3. A good 6K of Memory
4. A minimum of 1 to a maximum of 4 UBE on the system

2.2 Preliminary Requirements

It is expected that the module will have been tested on a GR or similar tester. This is to ensure that those areas that can not be thouroughly exercised by this program are working. These areas are:

1. Wrong Grant Error bit
2. No, No SACK time out Error bit
3. Wrong A lines Error bit
4. No Grant or not one Grant Error bit
5. No Interrupt SSYN Error bit
6. Inhibit Sack Logic.

In addition the passing of grants can not be tested if only one exercisor is present (see section 6.0). On those machines that don't have a parity trap (11/05, 11/20), the parity hardware is not checked. THE PARITY OPTION TEST (TEST 6) SHOULD BE DESELECTED BY SETTING SWITCH 5 FOR OTHER MACHINES WITHOUT PARITY MEMORY. ALSO, THE POWER DOWN TEST SHOULD NOT BE RUN ON THE 11/05.

2.3 Execution Time

For an error free, first pass run on an 11/45 with core memory, it takes approximately 15 seconds per UBE tested.

SEQ 0004

3.0 Starting Address

200 - for normal startup and restart
1100 - if halted in Interrupt test and wish to restart

4.0 Program Control and Operator Action

4.1

The paper tape is loaded using the standard procedure for ABS. tapes.

4.2

Load address 200

4.3

If the power down sequence is to be tested set SW4=1.

4.4

If more than one exercisor is present and it is desired to inhibit testing one or more of them, set the corresponding SW0,1,2,3=1. Switch 0 corresponds to the UBE which has the lowest address on the bus. Switch 1 to the next highest etc.. All UBE should not be inhibited. If this is done the program will trap to 4 after several end of passes. If all exercisors are to be tested SW0,1,2,3=0.

4.5

Start Test

5.0 Switch Options

THE USE OF THIS PROGRAM ON PROCESSORS HAVING A SOFTWARE SWITCH REGISTER NECESSITATES OPERATOR INTERACTION: THE OPERATOR MUST SET UP LOCATION 176 WITH THE SWITCH REGISTER VALUES DESIRED.

SW<15>=1 Halt on Error
SW<14>=1 Loop on Test

```
SW<13>=1    Inhibit Error Typouts
SW<12>=1    Inhibit Most Typeouts Except Error                    SEQ 0005
```

SW<11>=1 Inhibit Test Iterations
SW<10>=1 Bell on Error
SW<09>=1 Loop on Error
SW<05>=1 INHIBIT TEST 6
SW<04>=1 Test Power Down
SW<03>=1 Inhibit Test of UBE4
SW<02>=1 Inhibit Test of UBE3
SW<01>=1 Inhibit Test of UBE2
SW<00>=1 Inhibit Test of UBE1

5.1 SW<15>

The program halts on encountering an error after printing out the error message. Pressing 'continue' restores normal program operation.

5.2 SW<14>

The program loops on the subtest that is being executed when the switch is put on.

5.3 SW<13>

This switch inhibits all error typeouts

5.4 SW<12>

This switch inhibits most typeouts except error typeouts.

5.5 SW<11>

When one iterations of each test is inhibited.

5.6 SW<10>

The bell is rung upon encountering an error.

5.7 SW<09>

Upon finding an error, the program will cycle from the point of error to the previous scope statement (see sec. 8.2).

5.8 SW<05>

H 1

THE PARITY OPTION TEST (TEST 6) SHOULD BE DESELECTED BY SETTING SWITCH 5
FOR MACHINES WITHOUT PARITY MEMORY.

SEQ 0007

5.9 SW<04>

When set this switch enables the test of the power down sequence and the test that DCLO clears BECC, BEBA, BECR2 and BECR1 registers. This switch should not be set when running under ACT11 since a power down will cause an error statement from ACT.

5.10 SW<03>

When set this switch inhibits testing of the fourth UBE on the bus. The fourth exercisor is defined as the exercisor that responds to the fourth lowest address of the four exercisors. If there are less than four this switch has no effect on the program.

5.11 SW<02>

When set this switch inhibits test of that UBE with the third lowest address. If there are less than three, this switch has no effect on the program.

5.12 SW<01>

When set this switch inhibits test of that UBE with the second lowest address. If there are less than two, this switch has no effect on the program.

5.13 SW<00>

When set this switch inhibits testing the lowest address exercisor on the buss. If there is one exercisor, this switch should not be set.

6.0 Program Description

Upon start of the program, a map, called EMAP, of all the exercisors present is typed out in octal. Each bit set in the map corresponds to a UBE present. The least significant bit represents the UBE whose BEBD address is 770000. The second bit represents the UBE whose BEBD address is 770020 and so on.A maximum of 4 consecutive UBEs are allowed up to the maximum address of 770076. The addresses of the first UBE to be examined are then calculated and tests 1-37 are run.

The program then checks if more exercisors are to be tested up to a maximum of four. When these are done and if there were more than one UBE, the last test is executed. This tests the passing of grants

between the exercisors.

7.0 Error Reporting

Error calls are made via the EMT instruction. The lower byte of the instruction is encoded to indicate the error number. For example ERROR 1 would be (EMT+1) or 104001. Once an error instruction is executed, an error handler routine will then process the error call. The error message to be typed is determined from the item table at the beginning of the program. Item 1 corresponds to error 1 and so on. The item table contains a series of pointers to the message to be typed.

Every time an error occurs, the PC of the error call is typed out. This will tell the user the exact test where the error occurred. Many times other pertinent information is typed out as the contents of registers and bad addresses.

All messages refer to the UBE. For example, the message "DATI failed to set ready" means that the UBE when it did a DATI failed to set its ready.

It should be pointed out when trouble shooting a failing board, that the first error reported should be the first one fixed. This is because the nature of the hardware and software can cause additional, false or misleading error messages to appear after the first one. Since the tests build on one another and involve previously tested hardware, it will aid in the fault isolation to look up the tests previously run to know which hardware has been tested.Also,when multiple UBEs are being tested,a UBE can fail in such a way as to cause false error reports on a good board.This is especially true when the first failing UBE reports a "fatal error".Due to this,it is suggested that the first failing board reported should be repaired before proceeding to test the others.

8.0 Handlers and Common Routines

8.1 Trap Handler

This handler uses the trap instruction. The lower byte of the instruction is encoded differently for each of the different routines that use it. When a call for a routine is executed a trap occurs to the handler located at $TRAP. The handler then determines by looking at the lower byte which address to go to for servicing the call. The following routines use this handler:

1. TYPE - this routine is used to type ASCIZ messages.

2. TYPOC,TYPOS,TYPON - these routines are used to change a binary number to a 6 digit octal number and type it.

3. TYPDS - this routine converts a binary number to decimal number and types it.

8.2 Scope Handler

This handler is called via the 'IOT' trap. When 'scope' is executed an 'IOT' trap occurs to the memory location '$SCOPE'. Depending on the switch settings, the handler then decides to loop on test, loop on error etc. The scope statement that is located at the first instruction of the following test is the one that enables the desired action (looping etc.) for the present test.

8.3 Error Handler

This handler uses the 'EMT' trap. The lower byte of the instruction is encoded to indicate the error number. For example ERROR 1 would be (EMT+1) or 104001. Once an error instruction is executed the error handler determines the message to be typed. An item table at the beginning of the program contains pointers for each message to be typed. Each item corresponds to each error (Item 1 corresponds to error 1). The 'ERRTYP' routine then processes the table for the final error type out.

8.4 Trap Catcher

This is a series of instructions starting in location 0 to detect unexpected traps and interrupts to the trap and interrupt vector area of memory.

Each vector PC address is loaded with the address of the next location. The next location is loaded with a halt. Thus an illegal trap or interrupt will cause a halt at the trap PSW location plus 2.

Once a halt occurs, by examining the contents of the address pointed to by the stack, the value of the PC when the trap or interrupt occurred can be determined.

8.5 Power Down and Up Routines

When a power fail condition occurs, the contents of registers R0-R7 are saved on the stack. When the power returns, the same registers are restored.

8.6 CLRREG Routine

This subroutine will clear all the registers and error conditions of the uBE presently being tested.

8.7 RCATCH Routine

This routine restores the trap catcher to the vector area of the UBE presently being tested.

8.8 CRDY Routine

This routine checks for the ready bit to set from the UBE presently being tested. If ready fails to set in a time > 100 microseconds, the LSB of register R4 is set to a one.

8.9 DINT Routine

This routine is used to disregard interrupts from the UBE under test. It places the address of the next location in the UBE's vector area. The next location then contains an 'RTI' instruction.

8.10 RVEC Routine

This subroutine restores the vector area 0-56 from the stack and puts the trap catcher in the remaining locations.

8.11 TERRPC Routine

This routine is used any time an error occurs. It types out the PC of the error message, AND THE TEST NUMBER.

```
                        .TITLE  UNIBUS EXERCISOR MODULE DIAGNOSTIC
                        ;*COPYRIGHT (C) OCT 29,1974
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY WARREN SALTZ
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
000001                  $TN=1
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            INHIBIT MOST TYPEOUTS EXCEPT ERROR
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         5            WHEN SET, INHIBIT TEST 6
                        ;*         4            TEST POWER DOWN
                        ;*         3            INHIBIT TEST OF UBE 4
                        ;*         2            INHIBIT TEST OF UBE 3
                        ;*         1            INHIBIT TEST OF UBE 2
                        ;*         0            INHIBIT TETS OF UBE 1
                        .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 (BP.)
                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
                170000          DB=170000               ;DATA BUFFER OF LOWEST ADDRESS UBE
                                .SBTTL  TRAP CATCHER

                000000                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                000174                  .=174
000174  000000          DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000          SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  002632          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                001100          .=1100
001100  012737  000137  000200  RSTART: MOV     #000137,@#200   ;RESTART HERE IF HALTED IN INTERRUPT TEST
001106  012737  002632  000202          MOV     #START,@#202
001114  020627  001014          CMP R6,#1014            ;WAS VECTOR AREA DESTROYED IN INT. TEST?
001120  101002                  BHI B                   ;BRANCH IF NO
001122  004767  015166          JSR PC,RVEC             ;RESTORE VECTOR AREA
001126  000137  002632  B:      JMP     @#START         ;GO TO BEGINNING OF PROGRAM
                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                001132          $SVPC=.                 ;SAVE PC
                000046          .=46
000046  016134                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052          .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
                001132          .=$SVPC                 ;; RESTORE PC
```

```
                                .SBTTL  COMMON TAGS

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

                        001132          .=1132
001132                  $CMTAG:                         ;;START OF COMMON TAGS
001132  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001134     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001135     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001136  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001140  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001142  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001144  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001146     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001147     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001150  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001152  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001154  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001156  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001160  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001162  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001164  000000                  .WORD   0
001166     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001167     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001170  000000                  .WORD   0
001172  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001174  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001176  177560          $TKS:   177560                  ;;TTY KBD STATUS
001200  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001202  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001204  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001206     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001207     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001210     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001211     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001212  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001214  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001216  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001220  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001222  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001224  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001226  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001230  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001232  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001234  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001236  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001240  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001244     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001245     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001246  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**********************************************************************
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


001250                          $ERRTB:
                                ;ITEM1
001250  020676                          EM1
001252  000000                          0
001254  000000                          0
001256  000000                          0
                                ;ITEM2
001260  020760                          EM2
001262  021020                          DH2
001264  021046                          DT2
001266  000000                          0
                                ;ITEM3
001270  021054                          EM3
001272  021121                          DH3
001274  021130                          DT3
001276  000000                          0
                                ;ITEM 4
001300  021134                          EM4
001302  021202                          DH4
001304  021242                          DT4
001306  000000                          0
                                ;ITEM 5
001310  021252                          EM5
001312  021202                          DH4
001314  021242                          DT4
001316  000000                          0
                                ;ITEM 6
001320  021320                          EM6
001322  021202                          DH4
001324  021242                          DT4
001326  000000                          0
                                ;ITEM 7
001330  021370                          EM7
001332  021432                          DH7
001334  021510                          DT7
001336  000000                          0
                                ;ITEM 8
001340  021516                          EM8
001342  000000                          0
001344  000000                          0
001346  000000                          0
                                ;ITEM 9
```

```
   278  001350  021603                    EM9
   279  001352  000000                    0
   280  001354  000000                    0
   281  001356  000000                    0
   282                              ;ITEM 10
   283  001360  021644                    EM10
   284  001362  000000                    0
   285  001364  000000                    0
   286  001366  000000                    0
   287                              ;ITEM 11
   288  001370  021705                    EM11
   289  001372  000000                    0
   290  001374  000000                    0
   291  001376  000000                    0
   292                              ;ITEM 12
   293  001400  021751                    EM12
   294  001402  000000                    0
   295  001404  000000                    0
   296  001406  000000                    0
   297                              ;ITEM 13
   298  001410  000000                    0
   299  001412  000000                    0
   300  001414  000000                    0
   301  001416  000000                    0
   302                              ;ITEM 14
   303  001420  022016                    EM14
   304  001422  000000                    0
   305  001424  000000                    0
   306  001426  000000                    0
   307                              ;ITEM 15
   308  001430  022047                    EM15
   309  001432  022133                    DH15
   310  001434  021130                    DT3
   311  001436  000000                    0
   312                              ;ITEM 16
   313  001440  022155                    EM16
   314  001442  000000                    0
   315  001444  000000                    0
   316  001446  000000                    0
   317                              ;ITEM 17
   318  001450  022250                    EM17
   319  001452  022307                    DH17
   320  001454  021046                    DT2
   321  001456  000000                    0
   322                              ;ITEM 18
   323  001460  022334                    EM18
   324  001462  022423                    DH18
   325  001464  021130                    DT3
   326  001466  000000                    0
   327                              ;ITEM 19
   328  001470  022436                    EM19
   329  001472  022517                    DH19
   330  001474  021130                    DT3
   331  001476  000000                    0
   332                              ;ITEM 20
   333  001500  022540                    EM20
```

```
  334  001502  000000                    0
  335  001504  000000                    0
  336  001506  000000                    0
  337                              ;ITEM 21
  338  001510  022607                    EM21
  339  001512  000000                    0
  340  001514  000000                    0
  341  001516  000000                    0
  342                              ;ITEM 22
  343  001520  022645                    EM22
  344  001522  000000                    0
  345  001524  000000                    0
  346  001526  000000                    0
  347                              ;ITEM 23
  348  001530  022710                    EM23
  349  001532  000000                    0
  350  001534  000000                    0
  351  001536  000000                    0
  352                              ;ITEM 24
  353  001540  022754                    EM24
  354  001542  023023                    DH24
  355  001544  023100                    DT24
  356  001546  000000                    0
  357                              ;ITEM 25
  358  001550  023112                    EM25
  359  001552  023023                    DH24
  360  001554  023100                    DT24
  361  001556  000000                    0
  362                              ;ITEM 26
  363  001560  023152                    EM26
  364  001562  023023                    DH24
  365  001564  023100                    DT24
  366  001566  000000                    0
  367                              ;ITEM 27
  368  001570  023213                    EM27
  369  001572  023023                    DH24
  370  001574  023100                    DT24
  371  001576  000000                    0
  372                              ;ITEM 28
  373  001600  023253                    EM28
  374  001602  000000                    0
  375  001604  000000                    0
  376  001606  000000                    0
  377                              ;ITEM 29
  378  001610  023302                    EM29
  379  001612  000000                    0
  380  001614  000000                    0
  381  001616  000000                    0
  382                              ;ITEM 30
  383  001620  023331                    EM30
  384  001622  000000                    0
  385  001624  000000                    0
  386  001626  000000                    0
  387                              ;ITEM 31
  388  001630  023361                    EM31
  389  001632  000000                    0
```

```
  390  001634  000000                    0
  391  001636  000000                    0
  392                              ;ITEM 32
  393  001640  023411                    EM32
  394  001642  023023                    DH24
  395  001644  023100                    DT24
  396  001646  000000                    0
  397                              ;ITEM 33
  398  001650  023460                    EM33
  399  001652  023023                    DH24
  400  001654  023100                    DT24
  401  001656  000000                    0
  402                              ;ITEM 34
  403  001660  023515                    EM34
  404  001662  023565                    DH34
  405  001664  021130                    DT3
  406  001666  000000                    0
  407                              ;ITEM 35
  408  001670  023604                    EM35
  409  001672  023661                    DH35
  410  001674  021046                    DT2
  411  001676  000000                    0
  412                              ;ITEM 36
  413  001700  023707                    EM36
  414  001702  023756                    DH36
  415  001704  021130                    DT3
  416  001706  000000                    0
  417                              ;ITEM 37
  418  001710  023766                    EM37
  419  001712  023661                    DH35
  420  001714  021046                    DT2
  421  001716  000000                    0
  422                              ;ITEM 38
  423  001720  024025                    EM38
  424  001722  023661                    DH35
  425  001724  021046                    DT2
  426  001726  000000                    0
  427                              ;ITEM 39
  428  001730  024115                    EM39
  429  001732  000000                    0
  430  001734  000000                    0
  431  001736  000000                    0
  432                              ;ITEM 40
  433  001740  024173                    EM40
  434  001742  000000                    0
  435  001744  000000                    0
  436  001746  000000                    0
  437                              ;ITEM 41
  438  001750  024251                    EM41
  439  001752  000000                    0
  440  001754  000000                    0
  441  001756  000000                    0
  442                              ;ITEM 42
  443  001760  024313                    EM42
  444  001762  000000                    0
  445  001764  000000                    0
```

```
                                                    H  2
  446  001766  000000                    0
  447                                ;ITEM 43
  448  001770  024352                    EM43
  449  001772  024436                    DH43
  450  001774  021046                    DT2
  451  001776  000000                    0
  452                                ;ITEM 44
  453  002000  024467                    EM44
  454  002002  000000                    0
  455  002004  000000                    0
  456  002006  000000                    0
  457                                ;ITEM 45
  458  002010  024533                    EM45
  459  002012  000000                    0
  460  002014  000000                    0
  461  002016  000000                    0
  462                                ;ITEM 46
  463  002020  024560                    EM46
  464  002022  024630                    DH46
  465  002024  021242                    DT4
  466  002026  000000                    0
  467                                ;ITEM 47
  468  002030  024666                    EM47
  469  002032  024436                    DH43
  470  002034  021046                    DT2
  471  002036  000000                    0
  472                                ;ITEM 48
  473  002040  024666                    EM47
  474  002042  000000                    0
  475  002044  000000                    0
  476  002046  000000                    0
  477                                ;ITEM 49
  478  002050  024744                    EM49
  479  002052  000000                    0
  480  002054  000000                    0
  481  002056  000000                    0
  482                                ;ITEM 50
  483  002060  024744                    EM49
  484  002062  024436                    DH43
  485  002064  021046                    DT2
  486  002066  000000                    0
  487                                ;ITEM 51
  488  002070  025023                    EM51
  489  002072  000000                    0
  490  002074  000000                    0
  491  002076  000000                    0
  492                                ;ITEM 52
  493  002100  025054                    EM52
  494  002102  000000                    0
  495  002104  000000                    0
  496  002106  000000                    0
  497                                ;ITEM 53
  498  002110  025107                    EM53
  499  002112  000000                    0
  500  002114  000000                    0
  501  002116  000000                    0
```

```
                                                    I  2
502
503  002120  025161                    ;ITEM 54
                                              EM54
504  002122  000000                              0
505  002124  000000                              0
506  002126  000000                              0
507                                    ;ITEM 55
508  002130  024352                              EM43
509  002132  000000                              0
510  002134  000000                              0
511  002136  000000                              0
512                                    ;ITEM 56
513  002140  025424                              EM56
514  002142  000000                              0
515  002144  000000                              0
516  002146  000000                              0
517                                    ;ITEM 57
518  002150  025503                              EM57
519  002152  000000                              0
520  002154  000000                              0
521  002156  000000                              0
522                                    ;ITEM 58
523  002160  025533                              EM58
524  002162  022133                              DH15
525  002164  021130                              DT3
526  002166  000000                              0
527                                    ;ITEM 59
528  002170  025554                              EM59
529  002172  000000                              0
530  002174  000000                              0
531  002176  000000                              0
532                                    ;ITEM 60
533  002200  025622                              EM60
534  002202  000000                              0
535  002204  000000                              0
536  002206  000000                              0
537                                    ;ITEM 61
538  002210  025650                              EM61
539  002212  000000                              0
540  002214  000000                              0
541  002216  000000                              0
542                                    ;ITEM 62
543  002220  025677                              EM62
544  002222  000000                              0
545  002224  000000                              0
546  002226  000000                              0
547                                    ;ITEM 63
548  002230  025727                              EM63
549  002232  022133                              DH15
550  002234  021130                              DT3
551  002236  000000                              0
552                                    ;ITEM 64
553  002240  025757                              EM64
554  002242  000000                              0
555  002244  000000                              0
556  002246  000000                              0
557                                    ;ITEM 65
```

```
                                                    J 2
  558  002250  026055                    EM65
  559  002252  026135                    DH65
  560  002254  021130                    DT3
  561  002256  000000                    0
  562                                ;ITEM 66
  563  002260  026154                    EM66
  564  002262  000000                    0
  565  002264  000000                    0
  566  002266  000000                    0
  567                                ;ITEM 67
  568  002270  026202                    EM67
  569  002272  026135                    DH65
  570  002274  021130                    DT3
  571  002276  000000                    0
  572                                ;ITEM 68
  573  002300  000000                    0
  574  002302  026135                    DH65
  575  002304  021130                    DT3
  576  002306  000000                    0
  577                                ;ITEM 69
  578  002310  026244                    EM69
  579  002312  000000                    0
  580  002314  000000                    0
  581  002316  000000                    0
  582                                ;ITEM 70
  583  002320  026275                    EM70
  584  002322  000000                    0
  585  002324  000000                    0
  586  002326  000000                    0
  587                                ;ITEM 71
  588  002330  026356                    EM71
  589  002332  000000                    0
  590  002334  000000                    0
  591  002336  000000                    0
  592                                ;ITEM 72
  593  002340  026404                    EM72
  594  002342  000000                    0
  595  002344  000000                    0
  596  002346  000000                    0
  597                                ;ITEM 73
  598  002350  026434                    EM73
  599  002352  000000                    0
  600  002354  000000                    0
  601  002356  000000                    0
  602                                ;ITEM 74
  603  002360  026501                    EM74
  604  002362  000000                    0
  605  002364  000000                    0
  606  002366  000000                    0
  607                                ;ITEM 75
  608  002370  026523                    EM75
  609  002372  000000                    0
  610  002374  000000                    0
  611  002376  000000                    0
  612                                ;ITEM 76
  613  002400  026543                    EM76
```

```
   614  002402  000000                    0
   615  002404  000000                    0
   616  002406  000000                    0
   617                            ;ITEM 77
   618  002410  026570                    EM77
   619  002412  000000                    0
   620  002414  000000                    0
   621  002416  000000                    0
   622                            ;ITEM 78
   623  002420  026616                    EM78
   624  002422  021202                    DH4
   625  002424  021242                    DT4
   626  002426  000000                    0
   627                            ;ITEM 79
   628  002430  000000                    0
   629  002432  000000                    0
   630  002434  000000                    0
   631  002436  000000                    0
   632                            ;ITEM 80
   633  002440  026656                    EM80
   634  002442  000000                    0
   635  002444  000000                    0
   636  002446  000000                    0
   637                            ;ITEM 81
   638  002450  026720                    EM81
   639  002452  024630                    DH46
   640  002454  021242                    DT4
   641  002456  000000                    0
   642                            ;ITEM 82
   643  002460  026744                    EM82
   644  002462  000000                    0
   645  002464  000000                    0
   646  002466  000000                    0
   647                            ;ITEM 83
   648  002470  027011                    EM83
   649  002472  000000                    0
   650  002474  000000                    0
   651  002476  000000                    0
   652                            ;ITEM 84
   653  002500  027072                    EM84
   654  002502  000000                    0
   655  002504  000000                    0
   656  002506  000000                    0
   657  002510  000000  EMAP:    .WORD 0          ;MAP OF UBE PRESENT
   658  002512  000000  TMAP:    .WORD 0          ;TEMPORARY MAP
   659  002514  000000  SPTR:    .WORD 0          ;SWITCH POINTER
   660  002516  000000  BEBD:    .WORD 0          ;BEBD ADDRESS OF UBE UNDER TEST
   661  002520  000000  BECC:    .WORD 0          ;BECC ADDRESS OF UBE UNDER TEST
   662  002522  000000  BEBA:    .WORD 0          ;BEBA ADDRESS OF UBE UNDER TEST
   663  002524  000000  BECR1:   .WORD 0          ;BECR1 ADDRESS OF UBE UNDER TEST
   664  002526  000000  BECR2:   .WORD 0          ;BECR2 ADDRESS OF UBE UNDER TEST
   665  002530  000000  BERE:    .WORD 0          ;CLEAR ERROR ADDRESS OF UBE UNDER TEST
   666  002532  000000  INTVEC:  .WORD 0          ;INTERRUPT VECTOR ADDRESS OF UBE UNDER TEST
   667  002534  170014  BEGO:    .WORD 170014     ;GO ADDRESS
   668  002536  000000  BE1BD:   .WORD 0          ;BEBD ADDRESS OF FIRST UBE TESTED
   669  002540  000000  BE1CC:   .WORD 0          ;BECC ADDRESS OF FIRST UBE TESTED
```

```
   670  002542  000000                BE1BA:  .WORD 0                 ;BEBA ADDRESS OF FIRST UBE TESTED
   671  002544  000000                BE1CR1: .WORD 0                 ;BECR1 ADDRESS OF FIRST UBE TESTED
   672  002546  000000                BE1CR2: .WORD 0                 ;BECR2 ADDRESS OF FIRST UBE TESTED
   673  002550  000000                BE1RE:  .WORD 0                 ;CLEAR ERROR ADDRESS OF FIRST UBE TESTED
   674  002552  000000                BE1VEC: .WORD 0                 ;INTERRUPT VECTOR ADDRESS OF FIRST UBE TESTED
   675  002554  000000                BE2BD:  .WORD 0                 ;BEBD ADDRESS OF SECOND UBE TESTED
   676  002556  000000                BE2CC:  .WORD 0                 ;BECC ADDRESS OF SECOND UBE TESTED
   677  002560  000000                BE2BA:  .WORD 0                 ;BEBA ADDRESS OF SECOND UBE TESTED
   678  002562  000000                BE2CR1: .WORD 0                 ;BECR1 ADDRESS OF SECOND UBE TESTED
   679  002564  000000                BE2CR2: .WORD 0                 ;BECR2 ADDRESS OF SECOND UBE TESTED
   680  002566  000000                BE2RE:  .WORD 0                 ;CLEAR ERROR ADDRESS OF SECOND UBE TESTED
   681  002570  000000                BE2VEC: .WORD 0                 ;INTERRUPT VECTOR ADDRESS OF SECOND UBE TESTED
   682  002572  000000                BE3BD:  .WORD 0                 ;BEBD ADDRESS OF THIRD UBE TESTED
   683  002574  000000                BE3CC:  .WORD 0                 ;BECC ADDRESS OF THIRD UBE TESTED
   684  002576  000000                BE3BA:  .WORD 0                 ;BEBA ADDRESS OF THIRD UBE TESTED
   685  002600  000000                BE3CR1: .WORD 0                 ;BECR1 ADDRESS OF THIRD UBE TESTED
   686  002602  000000                BE3CR2: .WORD 0                 ;BECR2 ADDRESS OF THIRD UBE TESTED
   687  002604  000000                BE3RE:  .WORD 0                 ;CLEAR ERROR ADDRESS OF THIRD UBE TESTED
   688  002606  000000                BE3VEC: .WORD 0                 ;INTERRUPT VECTOR ADDRESS OF THIRD UBE TESTED
   689  002610  000000                BE4BD:  .WORD 0                 ;BEBD ADDRESS OF FOURTH UBE TESTED
   690  002612  000000                BE4CC:  .WORD 0                 ;BECC ADDRESS OF FOURTH UBE TESTED
   691  002614  000000                BE4BA:  .WORD 0                 ;BEBA ADDRESS OF FOURTH UBE TESTED
   692  002616  000000                BE4CR1: .WORD 0                 ;BECR1 ADDRESS OF FOURTH UBE TESTED
   693  002620  000000                BE4CR2: .WORD 0                 ;BECR2 ADDRESS OF FOURTH UBE TESTED
   694  002622  000000                BE4RE:  .WORD 0                 ;CLEAR ERROR ADDRESS OF FOURTH UBE TESTED
   695  002624  000000                BE4VEC: .WORD 0                 ;INTERRUPT VECTOR ADDRESS OF FOURTH UBE TESTED
   696  002626  000000                UCNT:   .WORD 0                 ;COUNT OF UBE TESTED
   697  002630  000000                NO:     .WORD 0                 ;INDEX NUMBER FOR ADDRESS OF 1,2,3,4 UBE
   698                                ;;**************************************************************
   699                                ;;**************************************************************
   700  002632                        START:
   701                                .SBTTL  INITIALIZE THE COMMON TAGS
   702                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
   703  002632  012706  001132                MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
   704  002636  005026                        CLR     (R6)+           ;;CLEAR MEMORY LOCATION
   705  002640  022706  001172                CMP     #SWR,R6 ;;DONE?
   706  002644  001374                        BNE     .-6             ;;LOOP BACK IF NO
   707  002646  012706  001100                MOV     #STACK,SP       ;;SETUP THE STACK POINTER
   708                                ;;INITIALIZE A FEW VECTORS
   709  002652  012737  016470  000020        MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
   710  002660  012737  000340  000022        MOV     #340,@#IOTVEC+2 ;;LEVEL 7
   711  002666  012737  016720  000030        MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
   712  002674  012737  000340  000032        MOV     #340,@#EMTVEC+2 ;;LEVEL 7
   713  002702  012737  020114  000034        MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
   714  002710  012737  000340  000036        MOV     #340,@#TRAPVEC+2;LEVEL 7
   715  002716  012737  020164  000024        MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
   716  002724  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;LEVEL 7
   717  002732  016767  013144  013134        MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
   718  002740  005067  176270                CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
   719  002744  005067  176266                CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
   720  002750  112767  000001  176171        MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
   721  002756  012767  002756  176154        MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
   722  002764  012767  002764  176150        MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
   723                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
   724                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
   725  002772  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
```

```
   726  002776  012737  003032  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
   727  003004  012767  177570  176160          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
   728  003012  012767  177570  176154          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
   729  003020  022777  177777  176144          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
   730  003026  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
   731                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
   732  003030  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
   733  003032  012716  003040          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
   734  003036  000002                          RTI
   735  003040  012767  000176  176124  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
   736  003046  012767  000174  176120          MOV     #DISPREG,DISPLAY
   737  003054  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
   738
   739  003060  032777  010000  176104          BIT #SW12,@SWR          ;INHIBIT TYPEOUTS?
   740  003066  001004                          BNE START1              ;BRANCH IF YES
   741  003070  104401  027665                  TYPE ,MSG16             ;UBE MODULE TEST
   742  003074  104401  027370                  TYPE ,MSG12             ;JUMPER W1 SHOULD BE IN TO PREVENT MULTIPLE SSYNS
   743  003100  005067  177404          START1: CLR EMAP                ;INIT. EMAP
   744  003104  012706  001100                  MOV     #STACK, SP      ;SETUP THE STACK POINTER
   745  003110  012767  000001  177376          MOV #1,SPTR             ;INITIALIZE SWITCH POINTER TO LOOK AT FIRST SWITCH
   746  003116  012767  002632  176016          MOV #START,$LPERR       ;SET UP RETURN FOR ERROR1
   747  003124  012737  003234  000004          MOV #MTRAP,@#4          ;SET UP MAP TRAP
   748  003132  012737  000340  000006          MOV #340,@#6            ;SET PSW PRIORITY=7
   749  003140  012701  170000                  MOV #DB,R1              ;DATA REG ADDR. OF FIRST REG
   750  003144  012700  000001                  MOV #1,R0               ;LD PTER
   751  003150  005711                  LOOP1:  TST (R1)                ;LOOK IF EXER. PRESENT,NO TRAPS
   752  003152  050067  177332                  BIS R0,EMAP             ;YES,INDIC. EXER. PRESENT
   753  003156  062701  000020          LOOP2:  ADD #20,R1              ;LOOK AT NEXT EXER. ADDR.
   754  003162  006100                          ROL R0                  ;UPDATE PTER
   755  003164  020027  000020                  CMP R0,#20              ;AT LAST UBE?
   756  003170  001367                          BNE LOOP1               ;BRANCH IF NOT AT LAST POSSIBLE EXER.
   757  003172  012737  000006  000004  A:      MOV #6,@#4              ;RESTORE TRAP CATCHER
   758  003200  005037  000006                  CLR @#6                 ;
   759  003204  032777  010000  175760          BIT #SW12,@SWR          ;'NHIBIT TYPEOUTS?
   760  003212  001007                          BNE 1$                  ;BRANCH IF YES
   761  003214  104401  020342                  TYPE ,MSG1              ;TYPE MAP
   762  003220  016746  177264                  MOV     EMAP,-(SP)      ;;SAVE EMAP FOR TYPEOUT
   763  003224  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
   764  003226  104401  001245                  TYPE ,$CRLF             ;
   765  003232  000415                  1$:     BR IADD                 ;GO CALC. ADDRESSES OF UBE
   766
   767  003234  022626                  MTRAP:  CMP     (SP)+,  (SP)+   ;RESTORE THE STACK
   768  003236  020027  000010                  CMP R0,#10              ;AT END OF UBE ADDRESS SPACE?
   769  003242  001345                          BNE LOOP2               ;NO LOOK AT NEXT EXER.
   770  003244  026727  177240  000000          CMP EMAP,#0             ;YES,IS MAP = 0?
   771  003252  001347                          BNE A                   ;NO,BRANCH TO A
   772  003254  104001                          ERROR  D1               ;NO RESPONSE TO REG ADDRESSES OR NO DEVICE PRESENT
   773  003256  004767  013146                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
   774  003262  000167  012560                  JMP $EOP                ;GO TO END OF TEST
   775                                  ;////////////////////////////////////////////////////////////////
   776                                  ;       ROUTINE TO CALCULATE ADDRESSES OF UBE TESTED
   777                                  ;////////////////////////////////////////////////////////////////
   778  003266  012767  167760  177222  IADD:   MOV #167760, BEBD         ;INITIALIZE BEBD
   779  003274  012767  167762  177216          MOV #167762, BECC       ;INITIALIZE BECC
   780  003302  012767  167764  177212          MOV #167764, BEBA       ;INITIALIZE BEBA
   781  003310  012767  167766  177206          MOV #167766, BECR1      ;INITIALIZE BECR1
```

```
  782  003316  012767  167776  177202          MOV #167776, BECR2      ;INITIALIZE BECR2
  783  003324  012767  167770  177176          MOV #167770, BERE       ;INITIALIZE BERE
  784  003332  012767  170014  177174          MOV #170014, BEGO       ;INITIALIZE BEGO
  785  003340  012767  000504  177164          MOV #504,    INTVEC     ;INITIALIZE INTERRUPT VECTOR
  786  003346  012700  002536                  MOV #BE1BD,R0           ;GET POINTER TO PERMANENT VECTOR AREA
  787  003352  005020                  1$:     CLR (R0)+               ;CLEAR PERMANENT VECTOR AREA
  788  003354  020027  002630                  CMP R0,#NO              ;ENTIRE AREA CLEARED?
  789  003360  001374                          BNE 1$                  ;BRANCH IF NO
  790  003362  012767  002536  177240          MOV #BE1BD,NO           ;INITIALIZE POINTER TO BE1BD
  791  003370  016767  177114  177114          MOV EMAP,TMAP           ;MOVE MAP TO WORK AREA
  792  003376  062767  000020  177112  ACALC:  ADD #20, BEBD           ;CALC. ADDR. OF BEBD TESTING
  793  003404  062767  000020  177106          ADD #20, BECC           ;CALC. ADDR. OF BECC TESTING
  794  003412  062767  000020  177102          ADD #20, BEBA           ;CALC. ADDR. OF BEBA TESTING
  795  003420  062767  000020  177076          ADD #20, BECR1          ;CALC. ADDR. OF BECR1 TESTING
  796  003426  062767  000020  177072          ADD #20, BECR2          ;CALC. ADDR. OF BECR2 TESTING
  797  003434  062767  000020  177066          ADD #20, BERE           ;CALC. ADDR. OF BERE TESTING
  798  003442  062767  000004  177062          ADD #4,  INTVEC         ;CALC. ADDR. OF INTERRUPT VECTOR
  799  003450  000241                          CLC                     ;INIT. CARRY
  800  003452  006267  177034                  ASR TMAP                ;LOOK FOR BIT INDICATING EXERCISOR
  801  003456  042767  100000  177026          BIC #100000,TMAP        ;CLEAR MSB IF SET
  802  003464  103405                          BCS C                   ;IF EXERCISOR PRESENT GO SEE IF TO BE TESTED
  803  003466  005767  177020                  TST TMAP                ;ANY EXERCISORS LEFT?
  804  003472  001341                          BNE ACALC               ;BRANCH IF MORE
  805  003474  000167  010736                  JMP LAST                ;GO TO LAST TEST
* 806  003500  032767  000020  177006  C:      BIT #20,SPTR            ;TESTED 4 UBE?
  807  003506  001402                          BEQ D                   ;BRANCH IF NO
  808  003510  000167  010722                  JMP LAST                ;GO TO LAST TEST
  809  003514  036777  176774  175450  D:      BIT SPTR,@SWR           ;SHOULD THIS UBE BE TESTED?
  810  003522  001403                          BEQ E                   ;BRANCH IF YES
  811  003524  006367  176764                  ASL SPTR                ;ROTATE POINTER TO NEXT SWITCH
  812  003530  000722                          BR ACALC                ;LOOK FOR NEXT UBE
  813  003532  006367  176756          E:      ASL SPTR                ;ROTATE POINTER TO NEXT SWITCH
  814  003536  005267  177064                  INC UCNT                ;UPDATE COUNT OF UBE TESTED
  815  003542  104401  027516                  TYPE ,MSG13             ;TESTING UBE WITH BEBD ADDRESS:
  816  003546  016746  176744                  MOV     BEBD,-(SP)      ;;SAVE BEBD FOR TYPEOUT
  817  003552  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  818  003554  104401  001245                  TYPE ,$CRLF
  819                                  ;////////////////////////////////////////////////////////////////////
  820                                  ;ROUTINE TO STORE TEMPORARY ADDRESS OF UBE TESTING IN PERMANENT LOC
  821                                  ;////////////////////////////////////////////////////////////////////
  822  003560  016701  177044                  MOV NO,R1               ;GET POINTER TO BE1BD
  823  003564  012700  002516                  MOV #BEBD,R0            ;GET POINTER FOR BEBD
  824  003570  012021          F:      MOV (R0)+,(R1)+         ;SAVE ADDRESSES
  825  003572  C20027  002534                  CMP R0,#BEGO            ;ALL SAVED?
  826  003576  001374                          BNE F                   ;BRANCH IF NO
  827  003600  062767  000016  177022          ADD #16,NO              ;UPDATE PTER TO NEXT UBE
  828
  829  003606  012767  003632  175324          MOV #FIRST,$LPADR       ;INIT. SCOPE WHEN MORE THAN 1 UBE
  830  003614  012767  003632  175320          MOV #FIRST,$LPERR       ;INIT. SCOPE WHEN MORE THAN 1 UBE
  831  003622  105067  175306                  CLRB $TSTNM             ;INIT. TEST NUMBER
  832  003626  000005                          RESET                   ;INIT. ALL UBE FOR LOOPS
  833
  834
  835                                  ;;**********************************************************
  836                                  ;*TEST 1         TEST ALL UBE REG CAN BE CLEARED
  837                                  ;*
```

```
                                        ;*R0 CONTAINS ADDRESS OF REG UNDER TEST
                                        ;*
                                        ;*IF THIS TEST FAILS, ALL FOLLOWING TESTS FOR THIS MODULE ARE ABORTED.
                                        ;;**************************************************************
003630  000004                  TST1:   SCOPE
003632  012706  001100          FIRST:  MOV #STACK,SP           ;RESTORE STACK
003636  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
003644  012737  004030  000004          MOV #STRAP,@#4          ;SET UP NSSYN TRAP
003652  012737  000340  000006          MOV #340,@#6            ;SET PSW PRIORTY =7
003660  012777  000000  176640          MOV #0,@BECR2           ;DO DATO TO CLEAR PB BIT IF SET
003666  005077  176636                  CLR @BERE               ;CLEAR ERROR CONDITIONS
003672  016700  176620                  MOV BEBD,R0             ;SETUP TO LOOK AT FIRST REG.
003676  005010                  T01L01: CLR (R0)                ;CLR UBE REG
003700  020067  176620                  CMP R0,BECR1            ;TESTING BECR1?
003704  001425                          BEQ T01L04              ;BRANCH IF YES
003706  005710                          TST (R0)                ;IS REG CLEARED?
003710  001421                          BEQ T01L02              ;BRANCH IF YES
003712  010067  175276          T01L03: MOV R0,$REG0            ;SAVE FAILING ADDRESS
003716  011067  175274                  MOV (R0),$REG1          ;SAVE BAD DATA
003722  104002                          ERROR  D2               ;FATAL ERROR:REG FAILED TO CLEAR
003724  020067  176576                  CMP R0,BECR2            ;DID BECR2 FAIL?
003730  001006                          BNE T01L06              ;BRANCH IF NO
003732  032777  020000  176566          BIT #20000,@BECR2       ;WAS CCOVF =1?
003740  001402                          BEQ T01L06              ;BRANCH IF NO
003742  104401  027560                  TYPE ,MSG14             ;DISREGARD BIT 13=1 OF BECR2
003746  004767  012456          T01L06: JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
003752  000433                          BR T01L05               ;RESTORE TRAP
003754  005720                  T01L02: TST (R0)+               ;INC ADDRESS
003756  000747                          BR T01L01               ;CONTINUE LOOP
003760  022777  000200  176536  T01L04: CMP #200,@BECR1         ;ALL BITS IN BECR1 0 EXCEPT RDY?
003766  001351                          BNE T01L03              ;BRANCH TO ERROR IF NO
003770  016700  176532                  MOV BECR2,R0            ;INDICATE LOOKING AT BECR2
003774  005077  176530                  CLR @BERE               ;RESET ERROR CONDITIONS
004000  005077  176522                  CLR @BECR2              ;CLEAR BECR2
004004  032777  157777  176514          BIT #157777,@BECR2      ;IS BECR2 =0 EXECPT CCOVF?
004012  001337                          BNE T01L03              ;NO, TYPE ADDRESS AND DATA ERROR
004014  012737  000006  000004          MOV #6,@#4              ;RESTORE TRAP CATCHER
004022  005037  000006                  CLR @#6
004026  000414                          BR      TST2            ;;GO TO NEXT TEST

004030  011667  175160          STRAP:  MOV (SP),$REG0          ;SAVE PC FROM STACK
004034  104003                          ERROR  D3               ;FATAL ERROR:CPU DID NOT RECEIVE SSYN
004036  004767  012366                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG

004042  012737  000006  000004  T01L05: MOV #6,@#4              ;RESTORE TRAP CATCHER
004050  005037  000006                  CLR @#6
004054  000167  010352                  JMP NUBE1               ;TEST NEXT UBE

                                        ;;**************************************************************
                                        ;*TEST 2        TEST BITS 1-6,8-14 OF BECR1 AND BITS 0-3,14 OF BECR2 CHANGE
                                        ;*
                                        ;*R2, R3 CONTAIN THE TRUE AND COMPLEMENT TEST DATA
                                        ;*R4 CONTAINS A POINTER TO THE REG ADDRESS BEING TESTED
                                        ;*R5 CONTAINS THE MASKED CONTENTS OF THE REG BEING TESTED
                                        ;*$TMP1 CONTAINS THE MASK FOR THE REG
                                        ;*
```

```
                                        ;*IF THIS TEST FAILS, ALL FOLLOWING TESTS FOR THIS MODULE ARE ABORTED
                                        ;;**************************************************************
004060  000004                TST2:   SCOPE
004062  012706  001100                MOV #STACK,SP           ;RESTORE STACK
004066  012737  000340  177776        MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
004074  012702  052652                MOV #52652,R2           ;SETUP TEST DATA BECR1
004100  012703  025324                MOV #25324,R3           ;SETUP COMP. TEST DATA BECR1
004104  012704  002524                MOV #BECR1,R4           ;LOAD ADDRESS PTER. FOR BECR1
004110  005077  176414                CLR @BERE               ;CLEAR ERROR CONDITIONS
004114  012767  177777  175104        MOV #177777,$TMP1       ;LOAD MASK TO LOOK AT ALL BECR1
004122  016705  175100        T02L03: MOV $TMP1,R5            ;LOAD R5 WITH MASK
004126  011400                        MOV (R4),R0             ;GET ADDRESS OF BECR TESTING
004130  010210                        MOV R2,(R0)             ;LOAD BECR WITH DATA
004132  011001                        MOV (R0),R1             ;GET CONTENTS OF BECR
004134  005101                        COM R1                  ;ONLY LOOK AT BITS
004136  040105                        BIC R1,R5               ;SET IN MASK =R5
004140  020502                        CMP R5,R2               ;DATA OK?
004142  001424                        BEQ T02L01              ;BRANCH IF YES
004144  011467  175044        T02L02: MOV (R4),$REG0          ;SAVE BECR ADDRESS
004150  011067  175042                MOV (R0),$REG1          ;SAVE BECR BAD DATA
004154  010267  175040                MOV R2,$REG2            ;SAVE GOOD DATA
004160  104006                        ERROR  D6               ;FATAL ERROR: CONTROL REG HELD WRONG DATA
004162  021467  176340                CMP (R4),BECR2          ;DID BECR2 FAIL?
004166  001006                        BNE T02L04              ;BRANCH IF NO
004170  032777  020000  176330        BIT #20000,@BECR2       ;WAS CCOVF=1?
004176  001402                        BEQ T02L04              ;BRANCH IF NO
004200  104401  027560                TYPE ,MSG14             ;DISREGARD BIT 13=1 OF BECR2
004204  004767  012220        T02L04: JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
004210  000167  010212                JMP NUBE                ;TEST NEXT UBE
004214  010302                T02L01: MOV R3,R2               ;XFER NEW TEST DATA
004216  010210                        MOV R2,(R0)             ;LOAD BECR WITH COMP.DATA
004220  011001                        MOV (R0),R1             ;GET CONTENTS OF BECR
004222  016705  175000                MOV $TMP1,R5            ;LOAD R5 WITH MASK
004226  005101                        COM R1                  ;ONLY LOOK AT BITS
004230  040105                        BIC R1,R5               ;SET IN MASK =R5
004232  020502                        CMP R5,R2               ;DATA OK?
004234  001343                        BNE T02L02              ;BRANCH IF NO
004236  012702  040012                MOV #40012,R2           ;SETUP TEST DATA BECR2
004242  012703  000005                MOV #5,R3               ;SETUP COMP. TEST DATA BECR2
004246  012704  002526                MOV #BECR2,R4           ;LOAD ADDRESS PTER. FOR BECR2
004252  012767  157777  174746        MOV #157777,$TMP1       ;HAVE MASK LOOK AT ALL BECR2 EXECPT CCOVF
004260  020067  176242                CMP R0,BECR2            ;TESTED BECR2?
004264  001316                        BNE T02L03              ;NO, BRANCH TO START TEST OF BECR2

                                        ;;**************************************************************
                                        ;*TEST 3         FLOAT A '1' THROUGH BEBD, BECC, BEBA
                                        ;*
                                        ;*R0 CONTAINS A POINTER TO THE REG ADDRESS BEING TESTED
                                        ;*R1 CONTAINS TEST DATA
                                        ;*
                                        ;*IF THIS TEST FAILS, ALL FOLLOWING TESTS FOR THIS MODULE ARE ABORTED
                                        ;;**************************************************************
004266  000004                TST3:   SCOPE
004270  012706  001100                MOV #STACK,SP           ;RESTORE STACK
004274  012737  000340  177776        MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
004302  012700  002516                MOV #BEBD,R0            ;GET BEBD ADDRESS PTER.
```

```
 950  004306  012701  000001          T03L04: MOV #1,R1                ;SETUP TEST DATA REG
 951  004312  010130                  T03L03: MOV R1,@(R0)+            ;PUT TEST DATA IN REG
 952  004314  025001                          CMP @-(R0),R1            ;TEST REG
 953  004316  001413                          BEQ T03L01               ;BRANCH IF OK
 954  004320  011067  174670                  MOV (R0),$REG0           ;SAVE FAILING REG ADDRESS
 955  004324  010167  174670                  MOV R1,$REG2             ;SAVE GOOD DATA
 956  004330  013067  174662                  MOV @(R0)+,$REG1         ;SAVE BAD DATA
 957  004334  104004                          ERROR  D4                ;FATAL ERROR:REG FAILED TO FLOAT A '1'
 958  004336  004767  012066                  JSR PC,TERRPC            ;TYPE PC OF ERROR MSG
 959  004342  000167  010060                  JMP NUBE                 ;TEST NEXT UBE
 960  004346  005701                  T03L01: TST R1                   ;TESTED ALL 16 BITS?
 961  004350  100402                          BMI T03L02               ;BRANCH IF YES
 962  004352  006301                          ASL R1                   ;TEST NEXT BIT
 963  004354  000756                          BR T03L03                ;CONTINUE LOOP
 964  004356  022067  176140          T03L02: CMP (R0)+,BEBA           ;TESTED LAST REG? ALSO UPDATE ADDR. PTER.
 965  004362  001351                          BNE T03L04               ;BRANCH IF REGS NOT TESTED
 966
 967                                  ;;*****************************************************************
 968                                  ;*TEST 4        FLOAT A '0' THROUGH BEBD,BECC,BEBA
 969                                  ;*
 970                                  ;*R0 CONTAINS A POINTER TO THE REG ADDRESS BEING TESTED
 971                                  ;*R1 CONTAINS TEST DATA
 972                                  ;*
 973                                  ;*IF THIS TEST FAILS, ALL FOLLOWING TESTS FOR THIS MODULE ARE ABORTED
 974                                  ;;*****************************************************************
 975  004364  000004                  TST4:   SCOPE
 976  004366  012706  001100                  MOV #STACK,SP            ;RESTORE STACK
 977  004372  012737  000340  177776          MOV #340,@#PSW           ;LOCK OUT INTERRUPTS
 978  004400  012700  002516                  MOV #BEBD,R0             ;GET BEBD ADDRESS PTER.
 979  004404  012701  177776          T04L04: MOV #177776,R1           ;SETUP TEST DATA REG
 980  004410  010130                  T04L03: MOV R1,@(R0)+            ;PUT TEST DATA IN REG
 981  004412  025001                          CMP@-(R0),R1             ;TEST REG
 982  004414  001413                          BEQ T04L01               ;BRANCH IF OK
 983  004416  011067  174572                  MOV (R0),$REG0           ;SAVE FAILING REG ADDRESS
 984  004422  010167  174572                  MOV R1,$REG2             ;SAVE GOOD DATA
 985  004426  013067  174564                  MOV @(R0)+,$REG1         ;SAVE BAD DATA
 986  004432  104005                          ERROR  D5                ;FATAL ERROR: REG FAILED TO FLOAT A '0'
 987  004434  004767  011770                  JSR PC,TERRPC            ;TYPE PC OF ERROR MSG
 988  004440  000167  007762                  JMP NUBE                 ;TEST NEXT UBE
 989  004444  005701                  T04L01: TST R1                   ;TESTED ALL 16 BITS?
 990  004446  100002                          BPL T04L02               ;BRANCH IF YES
 991  004450  006001                          ROR R1                   ;TEST NEXT BIT
 992  004452  000756                          BR T04L03                ;CONTINUE LOOP
 993  004454  022067  176042          T04L02: CMP(R0)+,BEBA            ;TESTED LAST REG? ALSO UPDATE ADDR. PTER.
 994  004460  001351                          BNE T04L04               ;BRANCH IF REG NOT TESTED
 995
 996                                  ;;*****************************************************************
 997                                  ;*TEST 5        TEST FOR DUAL ADDRESSING IN REGS
 998                                  ;*
 999                                  ;*THIS TEST CLEARS ALL REGS AND THEN WRITES INTO THE
1000                                  ;*REG BEING TESTED.  ALL OTHER REGS ARE THEN CHECKED IF THEY WERE
1001                                  ;*SIMULTANEOUSLY WRITTEN.  THIS IS THEN REPEATED FOR ALL REGS.
1002                                  ;*R0 CONTAINS ADDRESS OF REG BEING WRITTEN
1003                                  ;*R1 CONTAINS ADDRESS OF REG BEING EXAMINED
1004                                  ;*R2 CONTAINS MASK OF BITS TO BE LOOKED AT
1005                                  ;*
```

```
1006                                    ;*IF THIS TEST FAILS, ALL FOLLOWING TESTS FOR THIS MODULE ARE ABORTED
1007                                    ;;*****************************************************************
1008  004462  000004                    TST5:   SCOPE
1009  004464  012706  001100                    MOV #STACK,SP           ;RESTORE STACK
1010  004470  012737  000340  177776            MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
1011  004476  004767  011466                    JSR PC,CLRREG           ;CLEAR ALL REG
1012  004502  016700  176010                    MOV BEBD,R0             ;INITIALIZE TEST ADDRESS
1013  004506  016701  176004            T05L04: MOV BEBD,R1             ;INITIALIZE PTER.
1014  004512  012710  000002                    MOV #2,(R0)             ;LOAD TEST REG
1015  004516  012702  177777                    MOV #177777,R2          ;INITIALIZE MASK TO LOOK AT ALL BITS
1016  004522  030211                    T05L03: BIT R2,(R1)             ;IS DATA  IN REG =0?
1017  004524  001422                            BEQ T05L01              ;BRANCH IF DATA OK(=0)
1018  004526  020100                            CMP R1,R0               ;LOOKING AT REG LOADED?
1019  004530  001420                            BEQ T05L01              ;BRANCH IF YES (DATA OK)
1020  004532  020167  175766                    CMP R1,BECR1            ;LOOKING AT BECR1?
1021  004536  001411                            BEQ T05L07              ;BRANCH IF YES
1022  004540  010067  174450            T05L08: MOV R0,$REG0            ;ERROR; SAVE REG ADDRESS LOADED
1023  004544  010167  174446                    MOV R1,$REG1            ;SAVE REG ADDRESS EXAMINED
1024  004550  104007                            ERROR  D7               ;FATAL ERROR: DUAL ADDRESSING ERROR
1025  004552  004767  011652                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1026  004556  000167  007644                    JMP NUBE                ;TEST NEXT UBE
1027  004562  022777  000200  175734  T05L07: CMP #200,@BECR1          ;ALL BITS IN BECR1 0 EXCEPT RDY?
1028  004570  001363                            BNE T05L08              ;BRANCH IF NO
1029  004572  020167  175730            T05L01: CMP R1,BECR2            ;LOOKED AT BECR2?
1030  004576  001412                            BEQ T05L02              ;BRANCH IF YES
1031  004600  020167  175720                    CMP R1,BECR1            ;PTER UP TO BECR1?
1032  004604  001005                            BNE T05L06              ;NO, LOOK AT NEXT REG
1033  004606  016701  175714                    MOV BECR2,R1            ;NOW LOOK AT BECR2
1034  004612  012702  157777                    MOV #157777,R2          ;LOOK AT ALL BECR2 EXCEPT CCOVF
1035  004616  000741                            BR T05L03               ;CONTINUE LOOKING
1036  004620  005721                    T05L06: TST (R1)+               ;UPDATE PTER.
1037  004622  000737                            BR T05L03               ;LOOK AT NEXT REG.
1038  004624  004767  011340            T05L02: JSR PC,CLRREG           ;CLEAR ALL REG
1039  004630  020067  175672                    CMP R0,BECR2            ;LOADED AND TESTED BECR2?
1040  004634  001410                            BEQ     TST6            ;;BRANCH IF YES TO NEXT TEST
1041  004636  020067  175662                    CMP R0,BECR1            ;LOADED BECR1 WITH DATA YET?
1042  004642  001003                            BNE T05L05              ;BRANCH IF NO
1043  004644  016700  175656                    MOV BECR2,R0            ;YES, NOW LOAD BECR2
1044  004650  000716                            BR T05L04               ;CONTINUE LOOKING
1045  004652  005720                    T05L05: TST(R0)+                ;UPDATE ADDRESS OF REG LOADED
1046  004654  000714                            BR T05L04               ;TEST THIS REG
1047
1048                                    ;;*****************************************************************
1049                                    ;*TEST 6        TEST BUS PARITY BIT PB
1050                                    ;*
1051                                    ;*THIS TEST IS NOT RUN ON THOSE MACHINE
1052                                    ;*WITH NO PARITY TRAP (11/05, 11/20)
1053                                    ;*
1054                                    ;*FOR OTHER MACHINES, THIS TEST SHOULD BE DESELECTED IF THE
1055                                    ;*MEMORY PARITY OPTION IS NOT PRESENT OR NOT ENABLED, ELSE
1056                                    ;*AN ERROR WILL BE REPORTED ALTHOUGH HARDWARE IS FUNCTIONING
1057                                    ;*PROPERLY.
1058                                    ;*SW05=1        INHIBIT TEST 6 AND GO TO NEXT TEST
1059                                    ;;*****************************************************************
1060  004656  000004                    TST6:   SCOPE
1061  004660  012706  001100                    MOV #STACK,SP           ;RESTORE STACK
```

```
1062
1063                                    ;////////////////////////////////////////////////////////////////
1064  004664  032777  000040  174300            BIT     #SW05, @SWR     ;INHIBIT TEST 6?
1065  004672  001057                            BNE     TST7            ;GO TO NEXT TEST
1066                                    ;ROUTINE TO DETERMINE IF RUNNING UNDER 11/05 OR 11/20
1067                                    ;       IF 11/05 OR 11/20 BUSS PARITY TEST IS SKIPPED
1068                                    ;////////////////////////////////////////////////////////////////
1069  004674  012737  004770  000010            MOV #ITRAP,@#10         ;SET UP TO GO TO NEXT TEST IF ILLEGAL INST TRAP
1070  004702  012737  000340  000012            MOV #340,@#12
1071  004710  006700                            SXT R0                  ;IF INST TRAPS HAVE 11/05 OR 11/20
1072
1073  004712  012737  000340  177776            MOV #340,@#PSW          ;SET PSW PRIORITY=7
1074  004720  012737  004754  000114            MOV #PTRAP,@#114        ;SET UP PARITY TRAP
1075  004726  012737  000340  000116            MOV #340,@#116
1076  004734  012777  010000  175564            MOV #10000,@BECR2       ;ENABLE PB PARITY
1077  004742  005777  175560                    TST @BECR2              ;START PARITY TRAP
1078  004746  104010                            ERROR  D8               ;SETTING PB PARITY FAILED TO CAUSE CPU TO TRAP
1079  004750  004767  011454                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1080  004754  012737  000116  000114  PTRAP:    MOV #116,@#114          ;RESTORE TRAP CATCHER
1081  004762  005037  000116                    CLR @#116               ;RESTORE TRAP CATCHER
1082  004766  000411                            BR T06L01               ;SKIP MSG
1083  004770  032777  010000  174174  ITRAP:    BIT #SW12,@SWR          ;INHIBIT TYPEOUTS?
1084  004776  001005                            BNE T06L01              ;BRANCH IF YES
1085  005000  012767  000001  174226            MOV #1,$TIMES           ;DO 1 ITERATION WHEN TEST NOT NOT RUN
1086  005006  104401  020611                    TYPE ,MSG5              ;BUS PARITY NOT TESTED ON 11/05 OR 11/20 MACHINES
1087  005012  012737  000012  000010  T06L01:  MOV #12,@#10           ;RESTORE TRAP CATCHER
1088  005020  005037  000012                    CLR @#12                ;RESTORE TRAP CATCHER
1089  005024  012777  000000  175474            MOV #0,@BECR2           ;DO DATO TO CLEAR PB BIT
1090
1091                                    ;;**************************************************************
1092                                    ;*TEST 7        TEST GO WORKS, RDY SETS AND CLEARS, AND THE RELEASE BUS IMMEDIATE WORKS
1093                                    ;*
1094                                    ;*THE READY AND GO BIT ARE CHECKED USING A RELEASE
1095                                    ;*BUSS IMMEDIATE FUNCTION.  FALSE INTERRUPT ARE CHECKED FOR
1096                                    ;*
1097                                    ;*IF THE GO OR READY BITS FAIL, ALL FOLLOWING TESTS FOR THIS MODULE ARE ABORTED.
1098                                    ;;**************************************************************
1099  005032  000004                    TST7:   SCOPE
1100  005034  012706  001100                    MOV #STACK,SP           ;RESTORE STACK
1101  005040  012737  000340  177776            MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
1102  005046  004767  011116                    JSR PC,CLRREG           ;CLR ALL REG
1103  005052  012777  005172  175452            MOV #FINT1,@INTVEC      ;SET UP FOR FALSE INTERRUPT
1104  005060  016700  175446                    MOV INTVEC,R0           ;GET INTERRUPT VECTOR
1105  005064  012760  000340  000002            MOV #340,2(R0)          ;SET PSW PRIORITY=7
1106  005072  012777  006003  175424            MOV #6003,@BECR1        ;SET GO BIT AND DO RELEASE BUSS IMMEDIATE WITH BR4=1
1107  005100  032777  000200  175416            BIT #200,@BECR1         ;LOOK AT RDY BIT
1108  005106  001035                            BNE T07L08              ;BRANCH IF NOT CLEARED
1109  005110  005037  177776                    CLR @#PSW               ;ALLOW INTERRUPTS
1110  005114  005000                T07L07:     CLR R0                  ;INITIALIZE A COUNT TO WAIT FOR RDY=1
1111  005116  005200                T07L03:     INC R0                  ;UPDATE COUNT AND LOOP
1112  005120  022700  000011                    CMP #11,R0              ;TILL COUNT=10 OR RDY=1
1113  005124  001416                            BEQ T07L04              ;BRANCH IF RDY WAS NOT SET
1114  005126  105777  175372                    TSTB @BECR1             ;READY SET?
1115  005132  100371                            BPL T07L03              ;CONTINUE TO LOOK FOR RDY
1116  005134  032777  000001  175362            BIT #1,@BECR1           ;SEE IF GO BIT CLEARED
1117  005142  001426                            BEQ T07L05              ;PROCEED TO NEXT TEST IF YES
```

```
1118  005144  004767  011052                T07L04: ERROR  D12              ;FATAL ERROR: RDY BIT FAILED TO SET
1119  005150  104013                               JSR PC,TERRPC            ;TYPE PC OF ERROR MSG
1120  005152  004767  011252                       BR T07L06                ;ABORT UBE TEST
1121  005156  000167  007244
1122  005162  104014                       FINT1:  ERROR  D83               ;ERROR: FALSE INTERRUPT WHEN DO RELEASE BUSS IMMED.
1123  005164  004767  011240                       JSR PC,TERRPC            ;TYPE PC OF ERROR MSG
1124  005170  000407                               BR T07L07                ;NOW CHECK IF RDY=1 AND GO BIT=0
1125
1126  005172  104123                       T07L08: ERROR  D16               ;FATAL ERROR: RDY BIT FAILED TO CLEAR OR GO DID NOT SET
1127  005174  004767  011230                       JSR PC,TERRPC            ;TYPE PC OF ERROR MSG
1128  005200  000745                       T07L06: JSR PC,RCATCH            ;RESTORE TRAP CATCHER
1129                                               JMP NUBE                 ;TEST NEXT UBE
1130  005202  104020                       T07L05: JSR PC,RCATCH            ;RESTORE TRAP CATCHER
1131  005204  004767  011220
1132  005210  004767  011006
1133  005214  000167  007206
1134  005220  004767  010776
1135
1136                                       ;;*****************************************************************
1137                                       ;*TEST 10       TEST UBE CAN INTERRUPT ON ALL LEVELS & THE NO INT. SSYN BIT DOESN'T SET
1138                                       ;*
1139                                       ;*THE PSW PRIORITY IS FIRST SET EQUAL TO THE BR
1140                                       ;*LEVEL OF THE UBE.  ALL LEVELS ARE FIRST CHECKED
1141                                       ;*THIS WAY.  IF THE UBE FALSELY INTERRUPTS, A
1142                                       ;*SUBROUTINE, FINT3, WILL DETERMINE THE LEVEL IT
1143                                       ;INTERRUPTED.
1144                                       ;*AFTER THIS, THE UBE IS ALLOWED TO INTERRUPT BY
1145                                       ;*SETTING THE PSW PRIORITY ONE LEVEL BELOW THE BR.
1146                                       ;*ALL LEVELS ARE THEN CHECKED THIS WAY.  THE
1147                                       ;*PROPER INTERRUPT VECTOR IS TESTED FOR BY SETTING UP
1148                                       ;*THE ENTIRE VECTOR AREA 0-776 TO DETECT FOR WRONG
1149                                       ;*INTERRUPTS.
1150                                       ;*
1151                                       ;*NOTE:  IF THIS TEST IS HALTED IN THE MIDDLE
1152                                       ;*       AND IT IS DESIRED TO RESTART THE PROGRAM,
1153                                       ;*       THE PROGRAM SHOULD BE RESTARTED AT 1100 AND
1154                                       ;*       NOT 200.
1155                                       ;*****************************************************************
1156  005224  000004                       TST10:  SCOPE
1157  005226  012706  001100                       MOV #STACK,SP            ;RESTORE STACK
1158  005232  012737  000340  177776               MOV #340,@#PSW           ;LOCK OUT INTERRUPTS
1159  005240  004767  010724                       JSR PC,CLRREG            ;CLEAR ALL UBE REG
1160  005244  010667  173754                       MOV SP,$TMP0             ;SAVE STACK ADDRESS
1161  005250  005000                               CLR R0                   ;INIT. R0
1162  005252  012046                       T08L08: MOV (R0)+,-(SP)          ;SAVE VECTOR AREA 0-56
1163  005254  022700  000060                       CMP #60,R0               ;ALL SAVED?
1164  005260  001374                               BNE T08L08               ;BRANCH  IF NO
1165  005262  013746  000174                       MOV     @#174,  -(SP)    ;SAVE SOFTWARE SWR
1166  005266  013746  000176                       MOV     @#176,  -(SP)    ;
1167  005272  012737  000341  000002               MOV #341,@#2             ;SET UP VECTOR AREA TO DETECT WRONG INT. VECTORS
1168  005300  012700  000004                       MOV #4,R0                ;INITIALIZE ADDRESS REG
1169  005304  012720  005716               T08L01: MOV #WINT,(R0)+          ;PUT WRONG INTERRUPT PTER IN ALL VECTOR LOCATIONS
1170  005310  012720  000341                       MOV #341,(R0)+           ;PUT AN ODD PSW IN ALL PSW LOCATIONS
1171  005314  022700  001000                       CMP #1000,R0             ;AT END OF VECTOR AREA?
1172  005320  001371                               BNE T08L01               ;BRANCH IF NO
1173  005322  012777  005600  175202               MOV #FINT3,@INTVEC       ;SET UP UBE VECTOR AREA FOR FALSE INT.
```

```
 1174  005330  012767  000004  173656          MOV #4,$REG0            ;INDICATE DOING BR=4
 1175  005336  012767  000200  173652          MOV #200,$REG1          ;INDICATE PSW PRIORITY=4
 1176  005344  012777  000003  175152          MOV #3,@BECR1           ;HAVE UBE DO BR=4
 1177  005352  012737  000200  177776          MOV #200,@#PSW          ;SET PRIORITY=4
 1178  005360  000240                          NOP                     ;UBE SHOULD NOT INTERRUPT HERE
 1179  005362  012767  000005  173624          MOV #5,$REG0            ;INDICATE DOING BR=5
 1180  005370  012767  000240  173620          MOV #240,$REG1          ;INDICATE PSW PRIORITY=5
 1181  005376  012737  000240  177776          MOV #240,@#PSW          ;SET PRIORITY=5
 1182  005404  012777  000005  175112          MOV #5,@BECR1           ;HAVE UBE DO BR=5
 1183  005412  000240                          NOP                     ;UBE SHOULD NOT INTERRUPT HERE
 1184  005414  012767  000006  173572          MOV #6,$REG0            ;INDICATE DOING BR=6
 1185  005422  012767  000300  173566          MOV #300,$REG1          ;INDICATE PRIORITY=6
 1186  005430  012737  000300  177776          MOV #300,@#PSW          ;SET PRIORITY=6
 1187  005436  012777  000011  175060          MOV #11,@BECR1          ;HAVE UBE DO BR=6
 1188  005444  000240                          NOP                     ;UBE SHOULD NOT INTERRUPT HERE
 1189
 1190                                  ;NOW TEST UBE WILL INTERRUPT WITH PRIORITY ONE LEVEL BELOW BR
 1191
 1192  005446  012777  000002  175050          MOV #2,@BECR1           ;INITIALIZE UBE TO DO BR=4
 1193  005454  012767  000004  173532          MOV #4,$REG0            ;INITIALIZE INDICATOR FOR BR=4
 1194  005462  012767  000003  173526          MOV #3,$REG1            ;INITIALIZE INDICATOR FOR PRIORITY=3
 1195  005470  012777  005552  175034          MOV #T08L02,@INTVEC     ;SET RETURN ADDRESS WHEN GET PROPER INTERRUPT
 1196  005476  012737  000140  177776          MOV #140,@#PSW          ;INITIALIZE PSW PRIORITY=3
 1197  005504  000240                          NOP                     ;UBE SHOULD INTERRUPT HERE
 1198  005506  000413                          BR T08L09               ;BRANCH TO ERROR IF NO INT.
 1199  005510  005267  173500          T08L03: INC $REG0               ;INDICATE BR LEVEL DOING
 1200  005514  005267  173476                  INC $REG1               ;INDICATE PSW PRIORITY LEVEL DOING
 1201  005520  000257                          CCC                     ;CLEAR N,Z,V,C
 1202  005522  062737  000040  177776          ADD #40,@#PSW           ;SET PRIORITY LEVEL BELOW BR LEVEL
 1203  005530  005277  174770                  INC @BECR1              ;HAVE UBE DO BR 1 LEVEL ABOVE PRIORITY
 1204  005534  000240                          NOP                     ;UBE SHOULD INTERRUPT HERE
 1205  005536  004767  010552          T08L09: JSR PC,RVEC             ;RESTORE TRAP CATCHER AND HANDLER
 1206  005542  104021                          ERROR  D17              ;ERROR: UBE FAILED TO INTERRUPT
 1207  005544  004767  010660                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1208  005550  000472                          BR      T08L06          ;;BRANCH TO TEST NO INT. SSYN ERROR BIT
 1209  005552  022626          T08L02: CMP (SP)+,(SP)+         ;RESTORE STACK AFTER INTERRUPT
 1210  005554  032777  000020  174742          BIT #20,@BECR1          ;TESTED LAST BR?
 1211  005562  001063                          BNE     T08L07          ;;BRANCH IF YES TO TEST NO INT. SSYN ERROR BIT
 1212  005564  006377  174734                  ASL @BECR1              ;SHIFT BECR1 FOR NEXT BR LEVEL
 1213  005570  042777  000400  174726          BIC #400,@BECR1         ;CLEAR SHIFTED RDY BIT
 1214  005576  000744                          BR T08L03               ;GO TEST NEXT BR
 1215
 1216  005600  022626          FINT3:  CMP (SP)+,(SP)+         ;RESTORE STACK AFTER INTERRUPT
 1217  005602  004767  010506                  JSR PC,RVEC             ;RESTORE VECTOR AREA
 1218  005606  104022                          ERROR  D18              ;ERROR: UBE INT. WHEN PSW AT SAME PRIORITY LEVEL
 1219  005610  004767  010614                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1220  005614  032777  007740  174704          BIT #7740,@BECR2        ;SEE IF ERROR CONDITION OCCURRED IN BECR2
 1221  005622  001407                          BEQ T08L04              ;BRANCH IF NO
 1222  005624  017767  174676  173362          MOV @BECR2,$REG0        ;SAVE ERROR CONDITIONS
 1223  005632  104017                          ERROR  D15              ;ERROR: ERROR BITS IN BECR2 SET WHEN SHOULD=0
 1224  005634  004767  010570                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1225  005640  000445                          BR      TST11           ;;BRANCH TO NEXT TEST
 1226
 1227  005642  012777  005650  174662  T08L04: MOV #T08L05,@INTVEC     ;SET UP INTVEC TO FIND BR LEVEL UBE MADE
 1228  005650  012706  001100          T08L05: MOV #STACK,SP           ;RESTORE STACK
 1229  005654  062767  000040  173334          ADD #40,$REG1           ;RAISE PRIORITY LEVEL BY 1
```

```
 1230  005662  005267  173326                  INC $REG0               ;INDICATE NEW LEVEL OF PRIORITY
 1231  005666  016737  173324  177776          MOV $REG1,@#PSW         ;SET PSW PRIORITY
 1232  005674  005277  174624                  INC @BECR1              ;HAVE UBE INTERRUPT AGAIN
 1233  005700  000240                          NOP                     ;IF UBE INT. HERE, INCREMENT PRIORITY
 1234  005702  004767  010314                  JSR PC,RCATCH           ;RESTORE TRAP CATCHER
 1235  005706  104023                          ERROR  D19              ;ERROR: UBE FALSELY INTERRUPTED AT HIGHER LEVEL
 1236  005710  004767  010514                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1237  005714  000417                          BR      TST11           ;;BRANCH TO NEXT TEST
 1238
 1239  005716  022626                  WINT:   CMP (SP)+,(SP)+         ;RESTORE STACK AFTER INTERRUPT
 1240  005720  004767  010370                  JSR PC,RVEC             ;RESTORE VECTOR AREA
 1241  005724  104024                          ERROR  D20              ;ERROR: UBE INTERRUPTED TO WRONG VECTOR
 1242  005726  004767  010476                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1243  005732  004767  010356          T08L07: JSR PC,RVEC             ;RETURN VECTOR AREA WHEN FINISH BR TEST
 1244  005736  032777  004000  174562  T08L06: BIT #4000,@BECR2        ;WAS NO INT. SSYN ERROR BIT SET?
 1245  005744  001403                          BEQ     TST11           ;;BRANCH TO NEXT TEST IF NO
 1246  005746  104027                          ERROR  D23              ;ERROR: NO INT. SSYN BIT FALSELY SET
 1247  005750  004767  010454                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1248
 1249                                  ;;**************************************************************
 1250                                  ;*TEST 11       TEST THE NO,NO SACK ERROR BIT DOESN'T SET
 1251                                  ;*
 1252                                  ;*THE INHIBIT SACK BIT IS SET AND THE UBE IS TOLD TO
 1253                                  ;*DO A FUN. 3.THE NO,NO SACK ERROR BIT IS THEN
 1254                                  ;*CHECKED TO NOT HAVE SET.
 1255                                  ;;**************************************************************
 1256  005754  000004                  TST11:  SCOPE
 1257  005756  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
 1258  005762  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
 1259  005770  004767  010174                  JSR PC,CLRREG           ;CLEAR ALL UBE REGS.
 1260  005774  012777  000010  174524          MOV #10,@BECR2          ;ENABLE INH SACK IN BECR2
 1261  006002  012777  006003  174514          MOV #6003,@BECR1        ;DO  FUN 3 VIA  BR4
```

```
  1262  006010  005037  177776          CLR @#PSW               ;ALLOW INTERRUPTS
  1263  006014  000240                  NOP                     ;ALLOW UBE TO GET BUSS. CPU SHOULD TIME OUT
```

```
 1264
 1265  006016  005000                        CLR R0                  ;INIT COUNTER
 1266  006020  005200                1$:     INC R0                  ;INC COUNTER
 1267  006022  105700                        TSTB R0                 ;DELAY AT LEAST 41 USEC
 1268  006024  100375                        BPL 1$                  ;BRANCH IF NO
 1269  006026  032777  000200  174472        BIT #200,@BECR2         ;WAS NO, NO SACK BIT SET?
 1270  006034  001403                        BEQ  RTR                ;BRANCH IF NO
 1271  006036  104026                        ERROR  D22              ;ERROR: NO, NO SACK BIT FALSELY SET
```

```
1272  006040  004767  010364              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1273  006044  004767  010120      RTR:    JSR PC,CLRREG           ;CLEAR ALL UBE REG
1274
1275                                      ;;*****************************************************************
1276                                      ;*TEST 12       TEST DATI,DATIP,DATO,DATOB AND FUNCTION 1 WORK PROPERLY
1277                                      ;*
1278                                      ;*ALL DATA TRANSFERS ARE DONE VIA BR TRANSFERS.
1279                                      ;*EACH OPERATION (DATI, DATO, DATIP, DATOB) DOES ONE
1280                                      ;*TRANSFER AND THE DATA IS THEN CHECKED.
1281                                      ;*EACH TIME AN OPERATION IS STARTED THE READY
1282                                      ;*BIT IS TESTED BY THE SUBROUTINE 'RDYS' TO SEE IF IT SETS.
1283                                      ;;*****************************************************************
1284  006050  000004              TST12:  SCOPE
1285  006052  012706  001100              MOV #STACK,SP           ;RESTORE STACK
1286  006056  012737  000340  177776      MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
1287  006064  012767  052525  021716      MOV #052525,BUFF1       ;PUT TEST DATA IN BUFFER
1288  006072  004767  010072              JSR PC,CLRREG           ;CLEAR ALL UBE REG
1289  006076  012777  177777  174414      MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1290  006104  012777  030010  174410      MOV #BUFF1,@BEBA        ;LOAD UBE WITH BUFFER ADDRESS
1291  006112  012705  006620              MOV #ERR1,R5            ;INITIALIZE R5 FOR ERROR ADDRESS
1292  006116  012777  002003  174400      MOV #2003,@BECR1        ;HAVE UBE DO DATI VIA BR=4 AND FUNCTION 1
1293  006124  005037  177776              CLR @#PSW               ;ALLOW DATA XFER
1294  006130  004767  000434              JSR PC,RDYS             ;GO CHECK FOR RDY TO SET
1295  006134  022777  052525  174354      CMP #052525,@BEBD       ;IS DATA OK?
1296  006142  001421                      BEQ T10L01              ;GO TEST DATO IF YES
1297  006144  017767  174346  173042      MOV @BEBD,$REG0         ;SAVE (BEBD)
1298  006152  016767  021632  173036      MOV BUFF1,$REG1         ;SAVE MEM DATA
1299  006160  012767  030010  173032      MOV #BUFF1,$REG2        ;SAVE MEM ADDRESS
1300  006166  012767  052525  173026      MOV #52525,$REG3        ;SAVE CORRECT DATA
1301  006174  104030                      ERROR  D24              ;ERROR: DATI FAILED TO LOAD PROPER DATA
1302  006176  004767  010226              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1303  006202  000167  000450              JMP TSTA                ;GO TO NEXT TEST
1304  006206  004767  007756      T10L01: JSR PC,CLRREG           ;CLEAR UBE REG
1305  006212  005067  021572              CLR BUFF1               ;CLEAR TEST AREA
1306  006216  012777  177777  174274      MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1307  006224  012777  030010  174270      MOV #BUFF1,@BEBA        ;LOAD UBE WITH BUFFER ADDRESS
1308  006232  012777  052525  174256      MOV #052525,@BEBD       ;LOAD UBE WITH DATA
1309  006240  012705  006630              MOV #ERR2,R5            ;INITIALIZE R5 FOR ERROR ADDRESS
1310  006244  012777  003003  174252      MOV #3003,@BECR1        ;HAVE UBE DO DATO VIA BR=4 AND FUNCTION 1
1311  006252  004767  000312              JSR PC,RDYS             ;GO CHECK FOR RDY TO SET
1312  006256  022767  052525  021524      CMP #052525,BUFF1       ;WAS BUFFER LOADED PROPERLY?
1313  006264  001420                      BEQ     T10L02          ;GO TEST DATIP IF YES
1314  006266  017767  174224  172720      MOV @BEBD,$REG0         ;SAVE (BEBD)
1315  006274  016767  021510  172714      MOV BUFF1,$REG1         ;SAVE MEM DATA
1316  006302  012767  030010  172710      MOV #BUFF1,$REG2        ;SAVE MEM ADDRESS
1317  006310  012767  052525  172704      MOV #052525,$REG3       ;SAVE CORRECT DATA
1318  006316  104031                      ERROR  D25              ;ERROR: DATO FAILED TO LOAD PROPER DATA
1319  006320  004767  010104              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1320  006324  000554                      BR      TST13           ;;BRANCH TO NEXT TEST
1321
1322  006326  004767  007636      T10L02: JSR PC,CLRREG           ;CLEAR UBE REG
1323  006332  012767  052525  021450      MOV #052525,BUFF1       ;PUT TEST DATA IN BUFFER
1324  006340  012777  177777  174152      MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1325  006346  012777  030010  174146      MOV #BUFF1,@BEBA        ;LOAD UBE WITH BUFFER ADDRESS
1326  006354  012705  006640              MOV #ERR3,R5            ;INITIALIZE R5 FOR ERROR ADDRESS
1327  006360  012777  002403  174136      MOV #2403,@BECR1        ;HAVE UBE DO DATIP VIA BR=4 AND FUNCTION 1
```

```
 1328  006366  004767  000176              JSR PC,RDYS             ;GO CHECK FOR RDY SET
 1329  006372  022777  125252  174116      CMP #125252,@BEBD       ;HAS UBE SHIFTED DATA?
 1330  006400  001004                      BNE T10L06              ;BRANCH IF NO
 1331  006402  022767  125252  021400      CMP #125252,BUFF1       ;HAS MEM LOC BEEN SHIFTED?
 1332  006410  001420                      BEQ T10L03              ;GO TEST DATOB IF YES
 1333  006412  017767  174100  172574 T10L06: MOV @BEBD,$REG0       ;SAVE (BEBD)
 1334  006420  016767  021364  172570      MOV BUFF1,$REG1         ;SAVE MEM DATA
 1335  006426  012767  030010  172564      MOV #BUFF1,$REG2        ;SAVE MEM ADDRESS
 1336  006434  012767  125252  172560      MOV #125252,$REG3       ;SAVE CORRECT DATA
 1337  006442  104032                      ERROR  D26              ;ERROR: DATIP FAILED TO LOAD PROPER DATA
 1338  006444  004767  007760              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1339  006450  000502                      BR      TST13           ;;BRANCH TO NEXT TEST
 1340
 1341  006452  012767  000377  021330 T10L03: MOV #377,BUFF1        ;INITIALIZE BUFFER
 1342  006460  012705  006650              MOV #ERR4,R5            ;INITIALIZE R5 FOR ERROR ADDRESS
 1343  006464  012777  177400  174024      MOV #177400,@BEBD       ;LOAD HIGH BYTE OF UBE WITH 1'S
 1344  006472  012777  030011  174022      MOV #BUFF1+1,@BEBA      ;LOAD HIGH BYTE BUFF ADDR. INTO UBE
 1345  006500  012777  177777  174012      MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
 1346  006506  012777  003403  174010      MOV #3403,@BECR1        ;HAVE UBE DO DATOB VIA BR=4 AND FUNCTION 1
 1347  006514  004767  000050              JSR PC,RDYS             ;GO CHECK FOR RDY SET
 1348  006520  022767  177777  021262      CMP #177777,BUFF1       ;TEST IF DATOB DONE CORRECTLY
 1349  006526  001453                      BEQ     TST13           ;;BRANCH IF YES TO NEXT TEST
 1350
 1351  006530  017767  173762  172456      MOV @BEBD,$REG0         ;SAVE (BEBD)
 1352  006536  016767  021246  172452      MOV BUFF1,$REG1         ;SAVE NEW DATA
 1353  006544  012767  030010  172446      MOV #BUFF1,$REG2        ;SAVE MEM ADDRESS
 1354  006552  012767  177777  172442      MOV #177777,$REG3       ;SAVE CORRECT DATA
 1355  006560  104033                      ERROR  D27              ;ERROR: DATOB FAILED TO LOAD DATA PROPERLY
 1356  006562  004767  007642              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1357  006566  000433                      BR      TST13           ;;BRANCH TO NEXT TEST
 1358
 1359                          ;SUBROUTINE TO TEST IF RDY BIT SET
 1360
 1361  006570  005004              RDYS:   CLR R4                  ;INITIALIZE R4
 1362  006572  032777  000200  173724 T10L05: BIT #200,@BECR1       ;IS RDY SET?
 1363  006600  001006                      BNE T10L04              ;BRANCH IF YES
 1364  006602  005204                      INC R4                  ;UPDATE COUNT
 1365  006604  032704  000020              BIT #20,R4              ;COUNT=16?
 1366  006610  001770                      BEQ T10L05              ;IF NO, GO TEST RDY AGAIN
 1367  006612  005726                      TST (SP)+               ;RETURN STACK PTER
 1368  006614  000115                      JMP (R5)                ;GO INDICATE ERROR
 1369  006616  000207              T10L04: RTS PC                                  ;RETURN AND CHECK DATA
 1370  006620  104034              ERR1:   ERROR  D28              ;ERROR: DATI FAILED TO SET RDY
 1371  006622  004767  007602              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1372  006626  000413                      BR      TST13           ;;GO TO NEXT TEST
 1373  006630  104035              ERR2:   ERROR  D29              ;ERROR: DATO FAILED TO SET RDY
 1374  006632  004767  007572              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1375  006636  000407                      BR      TST13           ;;GO TO NEXT TEST
 1376  006640  104036              ERR3:   ERROR  D30              ;ERROR: DATIP FAILED TO SET RDY
 1377  006642  004767  007562              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1378  006646  000403                      BR      TST13           ;;GO TO NEXT TEST
 1379  006650  104037              ERR4:   ERROR  D31              ;ERROR: DATOB FAILED TO SET RDY
 1380  006652  004767  007552              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1381
 1382  006656                      TSTA:
 1383
```

```
1384                                    ;;*****************************************************************
1385                                    ;*TEST 13       TEST INHIBIT DATA SHIFT ON DATIP
1386                                    ;;*****************************************************************
1387  006656  000004                    TST13:  SCOPE
1388  006660  012706  001100                    MOV #STACK,SP           ;RESTORE STACK
1389  006664  004767  007300                    JSR PC,CLRREG           ;CLEAR UBE REG
1390  006670  005037  177776                    CLR @#PSW               ;ALLOW INTERRUPTS
1391  006674  012767  052525  021106            MOV #052525,BUFF1       ;PUT TEST DATA IN BUFFER
1392  006702  012777  177777  173610            MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1393  006710  012777  030010  173604            MOV #BUFF1,@BEBA        ;LOAD UBE WITH BUFFER ADDRESS
1394  006716  012777  022403  173600            MOV #22403,@BECR1       ;HAVE UBE DO DATIP WITH INH DATA SHIFT
1395  006724  004767  007314                    JSR PC,CRDY             ;CHECK FOR RDY BIT
1396  006730  005704                            TST R4                  ;DID RDY SET?
1397  006732  001404                            BEQ     T11L01          ;BRANCH IF YES
1398  006734  104036                            ERROR  D30              ;ERROR: DATIP FAILED TO SET RDY
1399  006736  004767  007466                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1400  006742  000427                            BR      TST14           ;;BRANCH TO NEXT TEST
1401  006744  022777  052525  173544  T11L01: CMP #052525,@BEBD       ;IS (BEBD) OK?
1402  006752  001004                            BNE T11L02              ;BRANCH IF NO
1403  006754  022767  052525  021026            CMP #052525,BUFF1       ;IS MEM OK?
1404  006762  001417                            BEQ     TST14           ;;BRANCH IF YES TO NEXT TEST
1405  006764  017767  173526  172222  T11L02: MOV @BEBD,$REG0        ;SAVE (BEBD)
1406  006772  016767  021012  172216            MOV BUFF1,$REG1         ;SAVE MEM DATA
1407  007000  012767  030010  172212            MOV #BUFF1,$REG2        ;SAVE MEM ADDRESS
1408  007006  012767  052525  172206            MOV #052525,$REG3       ;SAVE CORRECT DATA
1409  007014  104040                            ERROR  D32              ;ERROR: INH. DATA SHIFT ON DATIP FAILED
1410  007016  004767  007406                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1411
1412                                    ;;*****************************************************************
1413                                    ;*TEST 14       TEST DATOB ON DATIP
1414                                    ;;*****************************************************************
1415  007022  000004                    TST14:  SCOPE
1416  007024  012706  001100                    MOV #STACK,SP           ;RESTORE STACK
1417  007030  004767  007134                    JSR PC,CLRREG           ;CLEAR UBE REG
1418  007034  005037  177776                    CLR @#PSW               ;ALLOW INTERRUPTS
1419  007040  012767  177525  020742            MOV #177525,BUFF1       ;LOAD TEST DATA IN BUFFER
1420  007046  012777  030010  173446            MOV #BUFF1,@BEBA        ;LOAD UBE WITH LOW BYTE ADDRESS
1421  007054  012777  177777  173436            MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1422  007062  012777  042403  173434            MOV #42403,@BECR1       ;HAVE UBE DO DATOB ON DATIP
1423  007070  004767  007150                    JSR PC,CRDY             ;CHECK FOR RDY SET
1424  007074  022777  177253  173414            CMP #177253,@BEBD       ;CHECK (BEBD) OK
1425  007102  001004                            BNE T12L01              ;BRANCH IF NO
1426  007104  022767  177653  020676            CMP #177653,BUFF1       ;CHECK BUFFER OK
1427  007112  001417                            BEQ     TST15           ;;BRANCH IF YES TO NEXT TEST
1428  007114  017767  173376  172072  T12L01: MOV @BEBD,$REG0        ;SAVE (BEBD)
1429  007122  016767  020662  172066            MOV BUFF1,$REG1         ;SAVE MEM DATA
1430  007130  012767  030010  172062            MOV #BUFF1,$REG2        ;SAVE MEM ADDRESS
1431  007136  012767  177653  172056            MOV #177653,$REG3       ;SAVE CORRECT DATA
1432  007144  104041                            ERROR  D33              ;ERROR: DATOB ON DATIP FAILED
1433  007146  004767  007256                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1434
1435
1436                                    ;;*****************************************************************
1437                                    ;*TEST 15       TEST NO SSYN ERROR BIT WORKS & FUN A,B BITS RESET BY ERROR INTERRUPT
1438                                    ;*
1439                                    ;*A DATI NPR IS DONE TO A MEM LOC (760000) THAT RETURNS
```

```
                                        ;*NO SSYN.  THE NO SSYN ERROR BIT AND BIT 15 OF BECR1
                                        ;*ARE CHECKED TO SET.  THE ERROR INTERRUPT IS THEN TESTED.
                                        ;*AFTER THIS THE ERROR IS CLEARED BY THE CLEAR ERROR
                                        ;*ADDRESS.  FINALLY THE FUN A,B BITS (BITS 10,11 OF BECR1)
                                        ;*ARE EXAMINED TO SEE IF THEY RESET WHEN AN ERROR
                                        ;*INTERRUPT OCCURS.
                                        ;;*****************************************************************
007152  000004                  TST15:  SCOPE
007154  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
007160  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
007166  004767  006776                  JSR PC,CLRREG           ;CLEAR UBE REG
007172  012777  007320  173332          MOV #T23L01,@INTVEC     ;SET UP FOR INTERRUPTS
007200  012777  160000  173314          MOV #160000,@BEBA       ;LOAD UBE WITH TEST ADDRESS WHICH RETURNS NO SSYN
007206  012777  000003  173312          MOV #3,@BECR2           ;LOAD UBE WITH TEST ADDRESS WHICH RETURNS NO SSYN
007214  012777  177777  173276          MOV #177777,@BECC       ;HAVE UBE DO 1 CYCLE
007222  012777  002041  173274          MOV #2041,@BECR1        ;HAVE DATI NPR DONE
007230  004767  007010                  JSR PC,CRDY             ;WAIT TILL RDY SET
007234  032777  000400  173264          BIT #400,@BECR2         ;WAS NSSYN ERROR BIT SET?
007242  001004                          BNE T23L02              ;BRANCH IF YES
007244  104073                          ERROR  D59              ;ERROR: TEST OF NSSYN ERROR BIT FAILED
007246  104074                          ERROR  D60              ;TO SET BIT 8 OF BECR2
007250  004767  007154                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
007254  032777  100000  173242  T23L02: BIT #100000,@BECR1      ;WAS ERROR BIT SET?
007262  001004                          BNE T23L03              ;BRANCH IF YES
007264  104073                          ERROR  D59              ;ERROR: TEST OF NSSYN ERROR BIT FAILED
007266  104075                          ERROR  D61              ;TO SET BIT 15 OF BECR1
007270  004767  007134                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
007274  005037  177776          T23L03: CLR @#PSW               ;ALLOW UBE TO INTERRUPT
007300  000240                          NOP                     ;UBE SHOULD INTERRUPT HERE
007302  017767  173220  171704          MOV @BECR2,$REG0        ;SAVE BECR2
007310  104073                          ERROR  D59              ;ERROR: TEST OF NSSYN ERROR BIT FAILED
007312  104072                          ERROR  D58              ;TO INTERRUPT CPU
007314  004767  007110                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
007320  005077  173204          T23L01: CLR @BERE               ;CLEAR ERROR BITS
007324  032777  000400  173174          BIT #400,@BECR2         ;WAS NSSYN ERROR BIT CLEARED?
007332  001404                          BEQ T23L05              ;BRANCH IF YES TO TEST FUN A, B BITS
007334  104073                          ERROR  D59              ;ERROR: TEST OF NSSYN ERROR BIT FAILED
007336  104076                          ERROR  D62              ;TO CLEAR BIT 8 OF BECR2
007340  004767  007064                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
007344  032777  002000  173152  T23L05: BIT #2000,@BECR1        ;WAS FUN A BIT RESET?
007352  001404                          BEQ T23L06              ;BRANCH IF YES
007354  104073                          ERROR  D59              ;ERROR: TEST OF NSSYN ERROR BIT FAILED
007356  104016                          ERROR  D14              ;TO CLEAR BIT 10 OF BECR1
007360  004767  007044                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
007364  012777  160000  173130  T23L06: MOV #160000,@BEBA       ;LOAD UBE WITH TEST ADDRESS WHICH RETURNS NO SSYN
007372  012777  000003  173126          MOV #3,@BECR2           ;LOAD UBE WITH TEST ADDRESS WHICH RETURNS NO SSYN
007400  012777  177772  173112          MOV #177772,@BECC       ;DO 2 CYCLES
007406  012777  007426  173116          MOV #T23L07,@INTVEC     ;SET UP FOR INT
007414  012777  004041  173102          MOV #4041,@BECR1        ;HAVE UBE DO FUN2 DATI VIA NPR
007422  004767  006616                  JSR PC,CRDY             ;WAIT TILL RDY SETS
007426  032777  004000  173070  T23L07: BIT #4000,@BECR1        ;WAS FUN B BIT RESET
007434  001404                          BEQ T23L04              ;RESTORE TRAP
007436  104073                          ERROR  D59              ;ERROR: TEST OF NSSYN ERROR BIT FAILED
007440  104105                          ERROR  D69              ;TO CLEAR BIT 11 OF BECR1
007442  004767  006762                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
007446  004767  006550          T23L04: JSR PC,RCATCH           ;RESTORE TRAP
```

```
1496  007452  005077  173052                  CLR @BERE               ;CLEAR ALL ERROR CONDITIONS
1497
1498
1499                                  ;;*********************************************************************
1500                                  ;*TEST 16       TEST ADDRESS REG COUNTS BY 2 AND 1
1501                                  ;*
1502                                  ;*R0 CONTAINS THE TEST DATA
1503                                  ;;*********************************************************************
1504  007456  000004                  TST16:  SCOPE
1505  007460  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
1506  007464  004767  006500                  JSR PC,CLRREG           ;CLEAR UBE REGS
1507  007470  004767  006600                  JSR PC,DINT             ;DISREGARD UBE INTERRUPTS
1508  007474  005037  177776                  CLR @#PSW               ;ALLOW INTERRUPTS
1509  007500  012700  000002                  MOV #2,R0               ;INITIALIZE TEST COUNTER
1510  007504  012777  177777  173006  T14L02: MOV #177777,@BECC      ;HAVE UBE DO 1 XFER
1511  007512  012777  002003  173004          MOV #2003,@BECR1        ;HAVE UBE DO DATI
1512  007520  004767  006520                  JSR PC,CRDY             ;CHECK RDY SET
1513  007524  020077  172772                  CMP R0,@BEBA            ;IS ADDRESS CORRECT?
1514  007530  001057                          BNE T14L01              ;BRANCH TO ERROR IF NO
1515  007532  005200                          INC R0                  ;UPDATE R0
1516  007534  005200                          INC R0                  ;UPDATE R0
1517  007536  022700  000002                  CMP #2,R0               ;HAVE ALL ADRESSES BEEN TESTED?
1518  007542  001360                          BNE T14L02              ;LOOK AT NEXT ADDRESS IF NO
1519  007544  012777  177776  172750          MOV #177776,@BEBA       ;LOAD MAX ADDRESS IN LOWER 16 BITS UBE
1520  007552  012777  000003  172746          MOV #3,@BECR2           ;LOAD A16,A17 OF UBE WITH 1
1521  007560  012777  177777  172732          MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1522  007566  005277  172732                  INC @BECR1              ;HAVE UBE DO DATI
1523  007572  004767  006446                  JSR PC,CRDY             ;CHECK RDY SET
1524  007576  032777  000003  172722          BIT #3,@BECR2           ;TEST A16,A17=0
1525  007604  001042                          BNE T14L03              ;BRANCH TO ERROR IF NO
1526
1527                                  ;NOW TEST ADDRESS COUNTS BY 1
1528
1529  007606  012777  030011  172706          MOV #BUFF1+1,@BEBA      ;PUT ODD ADD OF BUFFER IN UBE
1530  007614  012777  177777  172676          MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
1531  007622  012777  003403  172674          MOV #3403,@BECR1        ;HAVE UBE DO DATOB
1532  007630  004767  006410                  JSR PC,CRDY             ;CHECK RDY
1533  007634  022777  030012  172660          CMP #BUFF1+2,@BEBA      ;DID ADDRESS UPDATE BY 1?
1534  007642  001434                          BEQ     T14L04          ;;BRANCHIF YES TO RESTORE TRAPS
1535  007644  017767  172652  171342          MOV @BEBA,$REG0         ;SAVE BAD ADDRESS
1536  007652  012767  030012  171336          MOV     #BUFF1+2,$REG1  ;SAVE GOOD ADDRESS
1537  007660  104045                          ERROR  D37              ;ERROR: BEBA DID NOT COUNT BY 1
1538  007662  004767  006542                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1539  007666  000422                          BR      T14L04          ;;GO TO RESTORE TRAPS
1540  007670  017767  172626  171316  T14L01: MOV @BEBA,$REG0         ;SAVE BAD ADDRESS
1541  007676  010067  171314                  MOV R0,$REG1            ;SAVE CORRECT ADDRESS
1542  007702  104043                          ERROR  D35              ;ERROR: BEBA DID NOT COUNT BY 2
1543  007704  004767  006520                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1544  007710  000411                          BR      T14L04          ;;GO TO RESTORE TRAPS
1545  007712  017700  172610          T14L03: MOV @BECR2,R0           ;GET ADDRESS BITS FROM UBE
1546  007716  042700  177774                  BIC #177774,R0          ;JUST LOOK AT A16,A17
1547  007722  010067  171266                  MOV R0,$REG0            ;SAVE ADDRESS
1548  007726  104044                          ERROR  D36              ;ERROR: BEBA BITS A16,A17 DID NOT COUNT = 0
1549  007730  004767  006474                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1550  007734  004767  006262          T14L04: JSR     PC,RCATCH       ;RESTORE TRAPS AND GO TO NEXT TEST
1551
```

```
                                ;;*****************************************************************
                                ;*TEST 17       TEST BUS ADDRESS BITS WILL CHANGE
                                ;*
                                ;*THE UBE BUS ADDRESS BITS ARE CHECKED TO
                                ;*SEE IF THEY CAN CHANGE FROM 0,1. SEVERAL DATIS
                                ;*ARE DONE FROM LOCATION 0, THE HIGHEST LOC IN THE FIRST
                                ;*8K AND FROM THE UBE SIMULTANEOUS GO ADDRESS.
                                ;;*****************************************************************
007740  000004                TST17:  SCOPE
007742  012706  001100                MOV #STACK,SP           ;RESTORE STACK
007746  004767  006216                JSR PC,CLRREG           ;CLEAR UBE REG
007752  004767  006316                JSR PC,DINT             ;DISREGARD INTERRUPTS
007756  005037  177776                CLR @#PSW               ;ALLOW DATA TRANSFERS

                                ;SIZE MEMORY FROM 4K TO 8K

007762  012737  010012  000004        MOV #T13L01,@#4         ;SET UP TIME OUT TRAP
007770  012700  017776                MOV #17776,R0           ;SET R0=LAST ADDRESS IN 1ST 4K OF MEM
007774  062700  004000        T13L02: ADD #4000,R0            ;UPDATE R0 TO NEXT 1K OF MEM
010000  005710                        TST (R0)                ;TEST IF 1K PRESENT. TIMES OUT IF NOT.
010002  022700  037776                CMP #37776,R0           ;AT 8K?
010006  001372                        BNE T13L02              ;LOOK AT NEXT 1K IF NOT
010010  000402                        BR T13L03
010012  162700  004000        T13L01: SUB #4000,R0            ;GET ADDRESS OF LAST 1K OF MEM PRESENT

010016  012737  000006  000004 T13L03: MOV #6,@#4              ;RESTORE TRAP
010024  011001                        MOV (R0),R1             ;SAVE CONTENTS OF LAST LOC IN FIRST 8K
010026  010010                        MOV R0,(R0)             ;PUT ADDRESS OF LOC IN MEM LOC
010030  012737  000000  000000        MOV #0,@#0              ;PUT 0 IN LOC 0
010036  012777  177777  172454        MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
010044  012777  002003  172452        MOV #2003,@BECR1        ;HAVE UBE DO DATI FROM MEM LOC 0
010052  004767  006166                JSR PC,CRDY             ;CHECK FOR RDY SET
010056  005777  172434                TST @BEBD               ;SEE IF UBE READ 0 FROM LOC 0
010062  001034                        BNE T13L04              ;BRANCH TO ERROR IF DATA NOT = 0
010064  010077  172432                MOV R0,@BEBA            ;HAVE UBE ADDRESS HIGHEST MEMORY IN 4K-8K LOCATIONS
010070  012777  177777  172422        MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
010076  005277  172422                INC @BECR1              ;HAVE UBE DO DATI FROM HIGHEST MEMORY IN 4K-8K LOCATIONS
010102  004767  006136                JSR PC,CRDY             ;CHECK FOR RDY SET
010106  020077  172404                CMP R0,@BEBD            ;DID UBE READ FROM PROPER LOCATION?
010112  001020                        BNE T13L04              ;BRANCH IF DATA  NOT = R0
010114  016777  172414  172400        MOV BEGO,@BEBA          ;HAVE UBE ADDRESS ITS GO ADDRESS
010122  012777  000003  172376        MOV #3,@BECR2           ;HAVE UBE ADDRESS ITS GO ADDRESS
010130  012777  177777  172362        MOV #177777,@BECC       ;HAVE UBE DO 1 XFER
010136  005277  172362                INC @BECR1              ;HAVE UBE DO DATI FROM GO ADDRESS
010142  004767  006076                JSR PC,CRDY             ;CHECK FOR RDY SET
010146  005777  172344                TST @BEBD               ;DID UBE READ PROPER LOCATION?
010152  001411                        BEQ T13L05              ;BRANCH IF YES
010154  017767  172342  171032 T13L04: MOV @BEBA,$REGO         ;GET ADDRESS+2 TRIED TO READ FROM
010162  162767  000002  171024        SUB #2,$REGO            ;CALC. ADDRESS TRIED TO READ FROM
010170  104042                        ERROR  D34              ;ERROR: UBE DID DATI FROM WRONG LOCATION
010172  004767  006232                JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
010176  004767  006020        T13L05: JSR PC,RCATCH           ;RESTORE TRAPS
010202  010110                        MOV R1,(R0)             ;RESTORE CONTENTS OF LAST LOC OF FIRST 8K

                                ;;*****************************************************************
                                ;*TEST 20       TEST CYCLE COUNT COUNTS BY 1 AND INCREMENTS WITH EACH INTERRUPT
```

```
1608                                    ;*
1609                                    ;*THE BECC REG IS CYCLED FROM 0 TO 177777 BY INTERRUPTING THE
1610                                    ;*CPU.  AFTER EACH INTERRUPT, THE REG IS COMPARED WITH R0 WHICH
1611                                    ;*CONTAINS THE PROPER DATA.
1612                                    ;;*****************************************************************
1613  010204  000004                   TST20:  SCOPE
1614  010206  012706  001100                   MOV #STACK,SP           ;RESTORE STACK
1615  010212  012737  000340  177776           MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
1616  010220  004767  005744                   JSR PC,CLRREG           ;CLEAR UBE REG
1617  010224  005000                           CLR R0                  ;INITIALIZE TEST COUNTER
1618  010226  012777  010250  172276           MOV #T15L01,@INTVEC     ;SET UP INT VECTOR AREA
1619  010234  012777  000003  172262  T15L03:  MOV #3,@BECR1           ;HAVE UBE INT.VIA BR=4
1620  010242  005037  177776                   CLR @#PSW               ;ALLOW INTERRUPTS
1621  010246  000240                           NOP                     ;UBE WILL INTERRUPT HERE
1622  010250  022626                T15L01:  CMP (SP)+,(SP)+         ;RESTORE STACK AFTER INTERRUPT
1623  010252  005200                           INC R0                  ;UPDATE TEST COUNTER
1624  010254  005700                           TST R0                  ;IS R0=0?
1625  010256  001423                           BEQ T15L02              ;RESTORE TRAPS IF YES
1626  010260  020077  172234                   CMP R0,@BECC            ;DID CYCLE COUNT UPDATE PROPERLY?
1627  010264  001763                           BEQ T15L03              ;INCREMENT BECC IF YES
1628  010266  017767  172226  170720           MOV @BECC,$REG0         ;SAVE BAD DATA
1629  010274  010067  170716                   MOV R0,$REG1            ;SAVE GOOD DATA
1630  010300  104046                           ERROR  D38              ;ERROR: INTERRUPT FAILED TO UPDATE BECC TO CORRECT VALUE
1631  010302  004767  006122                   JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1632  010306  012737  000340  177776           MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
1633  010314  012777  006003  172202           MOV #6003,@BECR1        ;HAVE UBE CYCLE SO IT SETS RDY
1634  010322  005037  177776                   CLR @#PSW               ;ALLOW UBE TO CYCLE
1635  010326  004767  005670          T15L02:  JSR PC,RCATCH           ;RESTORE TRAPS
1636
1637                                    ;;*****************************************************************
1638                                    ;*TEST 21       TEST INHIBIT INCREMENT OF BECC AND BEBA
1639                                    ;*
1640                                    ;*A DATI IS DONE VIA BR ARBITRATION AND THE BECC AND BEBA REGS
1641                                    ;*ARE CHECKED TO NOT INCREMENT.
1642                                    ;;*****************************************************************
1643  010332  000004                   TST21:  SCOPE
1644  010334  012706  001100                   MOV #STACK,SP           ;RESTORE STACK
1645  010340  012737  000340  177776           MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
1646  010346  004767  005616                   JSR PC,CLRREG           ;CLEAR UBE REG
1647  010352  012777  030010  172142           MOV #BUFF1,@BEBA        ;LOAD UBE WITH TEST ADDRESS
1648  010360  012777  177777  172132           MOV #177777,@BECC       ;LOAD TEST DATA INTO BECC
1649  010366  012767  000001  017414           MOV #1,BUFF1            ;SETUP BUFFER DATA
1650  010374  012777  000004  172124           MOV #4,@BECR2           ;HAVE UBE INH. INC. OF BECC AND BEBA
1651  010402  012777  002003  172114           MOV #2003,@BECR1        ;HAVE UBE DO DATI FROM BUFFER AREA
1652  010410  005037  177776                   CLR @#PSW               ;ALLOW DATA XFER
1653  010414  005777  172076          T16L01:  TST @BEBD               ;WAS DATA XFERED?
1654  010420  001775                           BEQ T16L01              ;WAIT TILL DATA IN BEBD
1655  010422  022777  177777  172070           CMP #177777,@BECC       ;CHECK BECC WAS NOT UPDATED
1656  010430  001010                           BNE T16L02              ;BRANCH IF WAS TO ERROR
1657  010432  022777  030010  172062           CMP #BUFF1,@BEBA        ;CHECK BEBA WAS NOT UPDATED
1658  010440  001407                           BEQ T16L03              ;BRANCH IF WAS NOT UPDATED
1659  010442  104047                           ERROR  D39              ;ERROR: BEBA INCREMENTED WHEN IT WAS INHIBITED
1660  010444  004767  005760                   JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1661  010450  000403                           BR T16L03
1662  010452  104050                T16L02:  ERROR  D40               ;ERROR: BECC INCREMENTED WHEN IT WAS INHIBITED
1663  010454  004767  005750                   JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
```

```
                                                T16L03: BIC #4,@BECR2           ;ALLOW BEBA AND BECC TO COUNT
                                                        JSR PC,CRDY             ;WAIT TILL UBE IS DONE

                                                ;;**************************************************************
                                                ;*TEST 22       TEST INT ENB AND CCOVF WORK AND THAT UBE WILL DO SEVERAL XFERS
                                                ;*
                                                ;*THE UBE IS SETUP TO DO 4 DATO XFERS VIA BR ARBITRATION AND
                                                ;*INTERRUPT WHEN DONE.  THE INTERRUPT IS CHECKED FOR
                                                ;*AND THEN A BUFFER AREA IS TESTED TO SEE IF EXACTLY
                                                ;*FOUR TRANSFERS WERE DONE.
                                                ;;**************************************************************
                                                TST22:  SCOPE
                                                        MOV #STACK,SP           ;RESTORE STACK
                                                        MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
                                                        JSR PC,CLRREG           ;CLEAR UBE REG
                                                        MOV #BUFF1,R0           ;GET BUFFER ADDRESS
                                                T17L01: CLR (R0)+               ;CLEAR BUFFER AREA
                                                        CMP R0,#BUFF1+20        ;AT END OF BUFFER?
                                                        BNE T17L01              ;BRANCH IF NO
                                                        MOV #377,@BEBD          ;SET UP XFER TEST DATA
                                                        MOV #BUFF1,@BEBA        ;LOAD UBE WITH BUFF ADDRESS
                                                        MOV #177774,@BECC       ;SET UBE TO DO 4 XFERS
                                                        MOV #T17L02,@INTVEC     ;SET UP INT VECTOR
                                                        MOV #3121,@BECR1        ;HAVE UBE DO DATO VIA BR=7 AND INTERRUPT ON DONE
                                                        CLR @#PSW               ;ALLOW XFERS
                                                        CLR R0                  ;INITIALIZE COUNT
                                                T17L03: INC R0                  ;UPDATE COUNT TO WAIT FOR INTERRUPT
                                                        CMP #1000,R0            ;WAITED LONG ENOUGH?
                                                        BNE T17L03              ;BRANCH IF NO
                                                        ERROR  D41              ;ERROR: UBE FAILED TO INT. ON DONE
                                                        JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
                                                        BR      T17L09          ;;GO RESTORE TRAPS
                                                T17L02: MOV #BUFF1,R0           ;GET START OF BUFFER
                                                T17L05: TST (R0)+               ;TEST FIRST 4 LOC WRITTEN
                                                        BEQ T17L04              ;BRANCH IF NOT WRITTEN TO ERROR
                                                        CMP #BUFF1+10,R0        ;LOOKED AT ALL WRITTEN LOCS.
                                                        BNE T17L05              ;BRANCH IF NO
                                                T17L06: TST (R0)+               ;TEST LAST 4 LOC WERE NOT WRITTEN
                                                        BNE T17L10              ;BRANCH TO ERROR IF WERE
                                                        CMP #BUFF1+20,R0        ;AT END OF BUFFER?
                                                        BNE T17L06              ;NO, LOOK AT NEXT LOCATION
                                                        BIT #100,@BECR1         ;YES, TEST INT. ON DONE BIT=0
                                                        BNE T17L07              ;BRANCH TO ERROR IF NOT=0
                                                        BIT #20000,@BECR2       ;TEST CCOVF=1
                                                        BEQ T17L08              ;BRANCH TO ERROR IF=0
                                                        MOV #6003,@BECR1        ;SET GO BIT TO SEE IF CCOVF IS RESET
                                                        BIT #20000,@BECR2       ;TEST CCOVF=0
                                                        BEQ     T17L09          ;;GO RESTORE TRAPS IF YES
                                                        ERROR  D42              ;ERROR: CCOVF NOT CLEARED BY GO
                                                        JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
                                                        BR      T17L09          ;;GO RESTORE TRAPS
                                                T17L04: TST -(R0)               ;CALC. LAST ADD. WRITTEN
                                                T17L10: TST -(R0)               ;CALC. LAST ADD. WRITTEN
                                                        CMP #BUFF1,R0           ;WERE ANY ADD. WRITTEN?
                                                        BLE T17L11              ;BRANCH IF YES
                                                        ERROR  D55              ;ERROR: UBE DID NOT DO DATO TO PROPER # OF LOC (4)
```

```
 1720  010724  004767  005500                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1721  010730  000420                          BR      T17L09          ;;GO RESTORE TRAPS
 1722  010732  012767  030010  170254  T17L11: MOV #BUFF1,$REG0        ;SAVE FIRST LOCATION WRITTEN
 1723  010740  010067  170252                  MOV R0,$REG1            ;SAVE LAST LOCATION WRITTEN
 1724  010744  104053                          ERROR  D43              ;ERROR: UBE DID NOT DO DATO TO PROPER # OF LOCATIONS (4)
 1725  010746  004767  005456                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1726  010752  000407                          BR      T17L09          ;;GO RESTORE TRAPS
 1727  010754  104054                  T17L07: ERROR  D44              ;ERROR: INT. ON DONE BIT NOT CLEARED
 1728  010756  004767  005446                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1729  010762  000403                          BR      T17L09          ;;GO RESTORE TRAPS
 1730  010764  104055                  T17L08: ERROR  D45              ;ERROR: CCOVF NOT SET
 1731  010766  004767  005436                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1732  010772  004767  005224          T17L09: JSR PC,RCATCH           ;RESTORE TRAPS
 1733
 1734                                  ;;**********************************************************************
 1735                                  ;*TEST 23       TEST DATA XFERS FROM BECC
 1736                                  ;*
 1737                                  ;*THE UBE IS SET UP TO DO 4 DATO XFERS VIA BR ARBITRATION FROM
 1738                                  ;*THE BECC REG TO A BUFFER AREA.  THE AREA IS THEN CHECKED.
 1739                                  ;;**********************************************************************
 1740  010776  000004                  TST23:  SCOPE
 1741  011000  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
 1742  011004  004767  005160                  JSR PC,CLRREG           ;CLEAR UBE REG
 1743  011010  005037  177776                  CLR @#PSW               ;ALLOW INTERRUPTS
 1744  011014  012700  030010                  MOV #BUFF1,R0           ;GET BUFFER ADDRESS
 1745  011020  005020                  T18L01: CLR (R0)+               ;CLEAR BUFFER AREA
 1746  011022  020027  030030                  CMP R0,#BUFF1+20        ;AT END OF BUFFER?
 1747  011026  001374                          BNE T18L01              ;BRANCH IF NO
 1748  011030  012777  030010  171464          MOV #BUFF1,@BEBA        ;LOAD STARTING ADDRESS INTO UBE
 1749  011036  012777  177774  171454          MOV #177774,@BECC       ;SETUP UBE TO DO 4 XFERS
 1750  011044  012777  013003  171452          MOV #13003,@BECR1       ;HAVE UBE DO 4 XFERS FROM BECC
 1751  011052  032777  000200  171444  T18L02: BIT #200,@BECR1         ;LOOK FOR RDY SET
 1752  011060  001774                          BEQ T18L02              ;BRANCH TILL SET
 1753  011062  012700  030010                  MOV #BUFF1,R0           ;GET BUFFER ADDRESS
 1754  011066  012701  177774                  MOV #177774,R1          ;INITIALIZE R1=TO FIRST DATA WORD
 1755  011072  022001                  T18L04: CMP (R0)+,R1            ;IS DATA OK?
 1756  011074  001005                          BNE T18L03              ;NO, GO TO ERROR
 1757  011076  005201                          INC R1                  ;UPDATE FOR NEXT DATA
 1758  011100  020027  030020                  CMP R0,#BUFF1+10        ;LOOKED AT ALL DATA?
 1759  011104  001372                          BNE T18L04              ;NO, LOOK AT NEXT WORD
 1760  011106  000412                          BR      TST24           ;;GO TO NEXT TEST
 1761
 1762  011110  005740                  T18L03: TST -(R0)               ;CALC. ADDRESS OF FAILURES
 1763  011112  010067  170076                  MOV R0,$REG0            ;SAVE ADDRESS
 1764  011116  011067  170074                  MOV (R0),$REG1          ;SAVE BAD DATA
 1765  011122  010167  170072                  MOV R1,$REG2            ;SAVE GOOD DATA
 1766  011126  104056                          ERROR  D46              ;ERROR: DATO FROM BECC NOT DONE PROPERLY
 1767  011130  004767  005274                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1768
 1769                                  ;;**********************************************************************
 1770                                  ;*TEST 24       TEST UBE CAN DO 2 XFERS PER BUS REQUEST
 1771                                  ;*
 1772                                  ;*THE UBE IS SET UP TO DO 2 DATO XFERS PER REQUEST VIA
 1773                                  ;*BR ARBITRATION.  THE CYCLE COUNT IS SET TO DO A TOTAL OF
 1774                                  ;*FOUR XFERS.  THE UBE IS TOLD TO GO. THE FIRST TIME
 1775                                  ;*THE CPU GETS THE BUS, AFTER THIS, THE PSW PRIORITY IS
```

```
 1776                                   ;*SET FOR 7 HOLDING OFF FURTHER UBE ACTION.  A BUFFER
 1777                                   ;*AREA IS THEN CHECKED THAT THE UBE DID EXACTLY 2 XFERS
 1778                                   ;*PER REQUEST.
 1779                                   ;;*****************************************************************
 1780  011134  000004                  TST24:  SCOPE
 1781  011136  012706  001100                  MOV #STACK,SP          ;RESTORE STACK
 1782  011142  012737  000340  177776          MOV #340,@#PSW         ;LOCK ON INTERRUPTS
 1783  011150  004767  005014                  JSR PC,CLRREG          ;CLEAR UBE REGS
 1784  011154  012700  030010                  MOV #BUFF1,R0          ;GET BUFFER ADDRESS
 1785  011160  005020                  T19L01: CLR (R0)+              ;CLEAR BUFFER AREA
 1786  011162  020027  030030                  CMP R0,#BUFF1+20       ;AT END OF BUFFER?
 1787  011166  001374                          BNE T19L01             ;CONTINUE TO CLEAR IF NO
 1788  011170  012777  030010  171324          MOV #BUFF1,@BEBA       ;LOAD BUFFER ADDRESS INTO UBE
 1789  011176  012777  177774  171314          MOV #177774,@BECC      ;SET UBE TO DO 4 XFERS
 1790  011204  012777  000377  171304          MOV #377,@BEBD         ;LOAD TEST DATA INTO UBE
 1791  011212  012777  005003  171304          MOV #5003,@BECR1       ;HAVE UBE DO 2 DATO/REQUEST VIA BR=4
 1792  011220  005037  177776                  CLR @#PSW              ;ALLOW UBE TO DO XFERS
 1793  011224  000240                          NOP                    ;UBE SHOULD DO 2 XFERS HERE
 1794  011226  012737  000340  177776          MOV #340,@#PSW         ;SET PRIORITY=7 TO STOP LAST 2 XFERS
 1795  011234  012700  030010                  MOV #BUFF1,R0          ;GET BUFF ADDRESS
 1796  011240  005720                  T19L03: TST (R0)+              ;WAS BUFF WRITTEN?
 1797  011242  001411                          BEQ T19L09             ;BRANCH TO ERROR IF NO
 1798  011244  020027  030014                  CMP R0,#BUFF1+4        ;LOOKED AT FIRST 2 LOCATIONS?
 1799  011250  001373                          BNE T19L03             ;BRANCH IF NO
 1800  011252  005720                  T19L04: TST (R0)+              ;TEST BUFF LOC NOT WRITTEN
 1801  011254  001005                          BNE T19L02             ;BRANCH TO ERROR IF WRITTEN
 1802  011256  020027  030020                  CMP R0,#BUFF1+10       ;LOOKED AT FOURTH LOC?
 1803  011262  001373                          BNE T19L04             ;BRANCH IF NO
 1804  011264  000421                          BR      T19L05         ;;GO TO END OF TEST
 1805  011266  005740                  T19L09: TST -(R0)              ;CALC LAST ADDRESS WRITTEN
 1806  011270  005740                  T19L02: TST -(R0)              ;CALC LAST ADDRESS WRITTEN
 1807  011272  022700  030010                  CMP #BUFF1,R0          ;WERE ANY ADDRESS WRITTEN?
 1808  011276  101404                          BLOS T19L07            ;BRANCH IF YES
 1809  011300  104060                          ERROR  D48             ;ERROR: UBE DID NOT DO 2 XFERS/REQUEST
 1810  011302  004767  005122                  JSR PC,TERRPC          ;TYPE PC OF ERROR MSG
 1811  011306  000410                          BR      T19L05         ;;GO TO END OF TEST
 1812  011310  012767  030010  167676  T19L07: MOV #BUFF1,$REG0       ;SAVE FIRST ADDRESS WRITTEN
 1813  011316  010067  167674                  MOV R0,$REG1           ;SAVE LAST ADDRESS WRITTEN
 1814  011322  104057                          ERROR  D47             ;ERROR: UBE DID NOT DO 2 XFERS FOR EACH REQUEST
 1815  011324  004767  005100                  JSR PC,TERRPC          ;TYPE PC OF ERROR MSG
 1816  011330  005037  177776          T19L05: CLR @#PSW              ;ALLOW LAST 2 XFERS
 1817  011334  000240                          NOP                    ;ALLOW UBE TO GET BUS
 1818  011336  004767  004702                  JSR PC,CRDY            ;WAIT TILL UBE FINISHES XFERS
 1819
 1820                                   ;;*****************************************************************
 1821                                   ;*TEST 25       TEST UBE CAN DO 2 DATIP XFERS PER REQUEST
 1822                                   ;*
 1823                                   ;*THE UBE IS SET UP TO DO 2 DATIP XFERS PER REQUEST VIA
 1824                                   ;*BR ARBITRATION.  THE CYCLE COUNT IS SET TO DO A TOTAL OF
 1825                                   ;*FOUR XFERS.  THE UBE IS TOLD TO GO. THE FIRST TIME
 1826                                   ;*THE CPU GETS THE BUS, AFTER THIS, THE PSW PRIORITY IS
 1827                                   ;*SET FOR 7 HOLDING OFF FURTHER UBE ACTION.  A BUFFER
 1828                                   ;*AREA IS THEN CHECKED THAT THE UBE DID EXACTLY 2 XFERS
 1829                                   ;*PER REQUEST.
 1830                                   ;;*****************************************************************
 1831  011342  000004                  TST25:  SCOPE
```

```
 1832  011344  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
 1833  011350  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
 1834  011356  004767  004606                  JSR PC,CLRREG           ;CLEAR UBE REG
 1835  011362  012700  030010                  MOV #BUFF1,R0           ;GET BUFFER ADDRESS
 1836  011366  012720  125252          T20L01: MOV #125252,(R0)+       ;LOAD TEST DATA
 1837  011372  020027  030020                  CMP R0,#BUFF1+10        ;LOADED FIRST 4 LOCS?
 1838  011376  001373                          BNE T20L01              ;BRANCH IF NO
 1839  011400  012777  030010  171114          MOV #BUFF1,@BEBA        ;LOAD BUFFER ADDRESS INTO UBE
 1840  011406  012777  177774  171104          MOV #177774,@BECC       ;SET UBE TO DO 4 CYCLES
 1841  011414  012777  004403  171102          MOV #4403,@BECR1        ;HAVE UBE DO 2 DATIP/REQUEST VIA BR-4
 1842  011422  005037  177776                  CLR @#PSW               ;ALLOW UBE TO DO CYCLES
 1843  011426  000240                          NOP                     ;UBE SHOULD DO XFERS HERE
 1844  011430  012737  000340  177776          MOV #340,@#PSW          ;SET PRIORITY = 7 TO STOP LAST 2 CYCLES
 1845  011436  012700  030010                  MOV #BUFF1,R0           ;GET BUFF ADDRESS
 1846  011442  022720  052525          T20L03: CMP #052525,(R0)+       ;TEST BUFF LOCS WRITTEN
 1847  011446  001012                          BNE T20L02              ;BRANCH TO ERROR IF NOT DONE PROPERLY
 1848  011450  022700  030014                  CMP #BUFF1+4,R0         ;LOOKED AT 2 WRITTEN LOCS?
 1849  011454  001372                          BNE T20L03              ;BRANCH IF NO
 1850  011456  022720  125252          T20L04: CMP #125252,(R0)+       ;TEST BUFF LOCS NOT WRITTEN
 1851  011462  001005                          BNE T20L08              ;BRANCH TO ERROR IF WRITTEN
 1852  011464  020027  030020                  CMP R0,#BUFF1+10        ;LOOKED AT FOURTH LOC?
 1853  011470  001372                          BNE T20L04              ;BRANCH IF NO
 1854  011472  000421                          BR      T20L05          ;;GO TO END OF TEST
 1855  011474  005740                  T20L02: TST -(R0)               ;CALC LAST ADDRESS WRITTEN
 1856  011476  005740                  T20L08: TST -(R0)               ;CALC LAST ADDRESS WRITTEN
 1857  011500  022700  030010                  CMP #BUFF1,R0           ;WERE ANY LOC WRITTEN?
 1858  011504  101404                          BLOS T20L06             ;BRANCH IF YES
 1859  011506  104061                          ERROR  D49              ;ERROR: DID NOT DO 2 DATIP/REQUEST
 1860  011510  004767  004714                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1861  011514  000410                          BR      T20L05          ;;GO TO END OF TEST
 1862  011516  012767  030010  167470  T20L06: MOV #BUFF1,$REG0        ;SAVE FIRST ADDRESS WRITTEN
 1863  011524  010067  167466                  MOV R0,$REG1            ;SAVE LAST ADDRESS WRITTEN
 1864  011530  104062                          ERROR  D50              ;ERROR: UBE DID NOT DO 2 DATIP/REQUEST
 1865  011532  004767  004672                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 1866  011536  005037  177776          T20L05: CLR @#PSW               ;ALLOW LAST 2 CYCLES
 1867  011542  000240                          NOP                     ;ALLOW UBE TO GET BUS
 1868  011544  004767  004474                  JSR PC,CRDY             ;WAIT FOR UBE TO FINISH XFERS
 1869
 1870                                  ;;*****************************************************************
 1871                                  ;*TEST 26       TEST DATA XFERS VIA NPR AND INT ON DONE WORK
 1872                                  ;*
 1873                                  ;*THIS IS THE FIRST TEST WHERE THE NPR IS EXERCISED.  ONE
 1874                                  ;*DATO NPR IS DONE TO A BUFFER AREA.  THE READY BIT IS
 1875                                  ;*THEN CHECKED FOR SETTING.  NEXT, THE SAME OPERATION IS
 1876                                  ;*REPEATED ONLY THE INTERRUPT ON DONE BIT IS SET.
 1877                                  ;*THE PROGRAM TESTS FOR THE INTERRUPT AND THEN EXAMINES
 1878                                  ;*THE BUFFER AREA TO SEE THAT ONLY ONE XFER WAS DONE.
 1879                                  ;;*****************************************************************
 1880  011550  000004                  TST26:  SCOPE
 1881  011552  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
 1882  011556  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
 1883  011564  004767  004400                  JSR PC,CLRREG           ;CLEAR UBE REG
 1884  011570  005067  016214                  CLR BUFF1               ;CLEAR BUFFER LOC
 1885  011574  012777  177777  170714          MOV #177777,@BEBD       ;LOAD UBE DATA REG WITH TEST DATA
 1886  011602  012777  030010  170712          MOV #BUFF1,@BEBA        ;LOAD UBE ADDRESS REG WITH BUFF ADD.
 1887  011610  012777  177777  170702          MOV #177777,@BECC       ;SET UBE TO DO 1 CYCLE
```

```
1888  011616  012777  003041  170700          MOV #3041,@BECR1        ;HAVE UBE DO DATO VIA NPR
1889  011624  000240                          NOP                     ;ALLOW UBE TO SET BUS
1890  011626  004767  004412                  JSR PC,CRDY             ;CHECK RDY SET
1891  011632  005704                          TST R4                  ;DID RDY SET?
1892  011634  001042                          BNE T21L01              ;BRANCH TO ERROR IF RDY DID NOT SET
1893  011636  005767  016146                  TST BUFF1               ;WAS DATO DONE?
1894  011642  001452                          BEQ T21L02              ;BRANCH TO ERROR IF NPR NOT DONE
1895  011644  005067  016140                  CLR BUFF1               ;CLEAR BUFF LOC
1896  011650  005067  016136                  CLR BUFF1+2             ;CLEAR BUFF LOC +2
1897  011654  012777  011724  170650          MOV #T21L03,@INTVEC     ;SET UP FOR INTERRUPT
1898  011662  012777  030010  170632          MOV #BUFF1,@BEBA        ;LOAD TEST ADDRESS
1899  011670  012777  177777  170622          MOV #177777,@BECC       ;SET UBE TO DO 1 CYCLE
1900  011676  012777  003143  170620          MOV #3143,@BECR1        ;HAVE UBE DO DATO NPR AND INT WHEN DONE VIA BR=4
1901  011704  005037  177776                  CLR @#PSW               ;ALLOW UBE TO INTERRUPT
1902  011710  004767  004330                  JSR PC,CRDY             ;WAIT FOR INT. OR RDY TO SET
1903  011714  104065                          ERROR  D53              ;ERROR: UBE DID NOT INT WHEN NPR DONE
1904  011716  004767  004506                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1905  011722  000425                          BR T21L04               ;RESTORE TRAPS
1906  011724  005767  016062          T21L03: TST BUFF1+2             ;DID NPR WRITE MORE THAN 1 LOC?
1907  011730  001422                          BEQ     T21L04          ;;GO TO END OF TEST
1908  011732  104066                          ERROR  D54              ;ERROR: UBE WROTE 2 LOC WHEN 1 NPR AND INT DONE
1909  011734  004767  004470                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1910  011740  000416                          BR T21L04               ;RESTORE TRAPS
1911  011742  104064                  T21L01: ERROR  D52              ;ERROR: NPR DID NOT SET RDY
1912  011744  004767  004460                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1913  011750  012777  006003  170546          MOV #6003,@BECR1        ;HAVE UBE SET ITS RDY
1914  011756  005037  177776                  CLR @#PSW               ;
1915  011762  004767  004256                  JSR PC,CRDY             ;WAIT TILL SET
1916  011766  000403                          BR T21L04               ;RESTORE TRAPS
1917  011770  104063                  T21L02: ERROR  D51              ;ERROR: NPR DATO NOT DONE
1918  011772  004767  004432                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1919  011776  004767  004220          T21L04: JSR PC,RCATCH           ;RESTORE TRAPS
1920
1921                                  ;;**************************************************************
1922                                  ;*TEST 27       TEST UBE WILL NOT INTERRUPT DURING AN NPR AND GO BIT SETS
1923                                  ;*
1924                                  ;*IF THIS TEST FAILS AND THE UBE DOES INTERRUPT AFTER
1925                                  ;*TRYING TO DO AN NPR, THE CPU WILL GO DOWN
1926                                  ;;**************************************************************
1927  012002  000004                  TST27:  SCOPE
1928  012004  012767  000001  167222          MOV     #1,$TIMES       ;;DO 1 ITERATION
1929  012012  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
1930  012016  004767  004146                  JSR PC,CLRREG           ;CLEAR UBE REG
1931  012022  032777  010000  167142          BIT #SW12,@SWR          ;INHIBIT TYPEOUTS?
1932  012030  001002                          BNE 1$                  ;BRANCH IF YES
1933  012032  104401  025257                  TYPE ,MSG3              ;TESTING UBE WILL NOT INTERRUPT
1934                                          ;DURING NPR. IF DOES, CPU WILL GO DOWN
1935  012036  012777  177777  170454  1$:     MOV #177777,@BECC       ;SET UBE TO DO 1 CYCLE
1936  012044  012777  000043  170452          MOV #0043,@BECR1        ;HAVE UBE DO DATI NPR AND INT. (FUN.=0)
1937  012052  005037  177776                  CLR @#PSW
1938  012056  000240                          NOP                     ;UBE SHOULD NOT GET BUSS HERE
1939  012060  032777  000001  170436          BIT #1,@BECR1           ;IS GO BIT SET?
1940  012066  001003                          BNE T22L01              ;BRANCH IF YES
1941  012070  104011                          ERROR  D9               ;ERROR: GO BIT FAILED TO LOAD '1'
1942  012072  004767  004332                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
1943  012076  005077  170422          T22L01: CLR @BECR1              ;RESET GO BIT, NPR AND INTERRUPT
```

```
                012102  032777  010000  167062          BIT #SW12,@SWR          ;INHIBIT TYPEOUTS?
                012110  001002                          BNE     TST30           ;;BRANCH IF YES
                012112  104401  025405                  TYPE ,MSG4              ;EXITING TEST
                                                        ;;**************************************************************
                                                        ;*TEST 30       TEST WRONG A LINE ERROR BIT DOES NOT SET
                                                        ;*
                                                        ;*A DATI NPR IS DONE FROM THE UBE GO ADDRESS
                                                        ;*THE ERROR BIT IS TESTED NOT TO HAVE SET AND NOT TO HAVE INTERRUPTED.
                                                        ;*THE ADDRESS BITS 14,15,16,17 ARE NEXT TESTED SEPARATELY
                                                        ;*AND THE ERROR BIT IS CHECKED NOT TO HAVE SET.
                                                        ;;**************************************************************
                012116  000004                  TST30:  SCOPE
                012120  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
                012124  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
                012132  004767  004032                  JSR PC,CLRREG           ;CLEAR UBE REGS
                012136  016777  170372  170356          MOV BEGO,@BEBA          ;HAVE UBE ADDRESS ITS GO ADDRESS
                012144  012777  000003  170354          MOV #3,@BECR2           ;HAVE UBE ADDRESS ITS GO ADDRESS
                012152  012777  177777  170340          MOV #177777,@BECC       ;SET UP TO DO 1 CYCLE
                012160  012777  012230  170344          MOV #T24L01,@INTVEC     ;SET UP FOR INT.
                012166  012777  002041  170330          MOV #2041,@BECR1        ;HAVE DATI NPR DONE FROM GO ADDRESS
                012174  004767  004044                  JSR PC,CRDY             ;CHECK FOR RDY SET
                012200  032777  001000  170320          BIT #1000,@BECR2        ;WAS ADDRESS ERROR SET?
                012206  001404                          BEQ T24L02              ;BRANCH IF NO
                012210  104070                          ERROR  D56              ;ERROR: TEST OF WRONG A LINES ERROR BIT FAILED
                012212  104071                          ERROR  D57              ;BECR2 BIT 9 FALSELY SET
                012214  004767  004210                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
                012220  005037  177776          T24L02: CLR @#PSW               ;ALLOW ANY INTERRUPTS
                012224  000240                          NOP                     ;UBE SHOULD NOT INTERRUPT HERE
                012226  000410                          BR T24L06               ;GO TEST INDIVIDUAL ADDRESS BITS
                012230  017767  170272  166756  T24L01: MOV @BECR2,$REGO        ;SAVE BECR2
                012236  104070                          ERROR  D56              ;ERROR:TEST OF WRONG A LINES ERROR BIT FAILED
                012240  104077                          ERROR  D63              ;FALSELY INTERRUPTED CPU
                012242  004767  004162                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
                012246  000447                          BR T24L03               ;GO RESTORE TRAP
                012250  004767  004020          T24L06: JSR PC,DINT             ;DISREGARD INTERRUPTS
                012254  005077  170242                  CLR @BEBA               ;CLEAR ADDRESS 0-15
                012260  012777  000001  170240          MOV #1,@BECR2           ;TEST ADDRESS 16
                012266  012777  177777  170224  T24L05: MOV #177777,@BECC      ;DO 1 CYCLE
                012274  062777  040000  170220          ADD #40000,@BEBA        ;TEST NEXT ADDRESS
                012302  032777  140000  170212          BIT #140000,@BEBA       ;HAVE ADDRESS BITS 14,15 BEEN EXERCISED?
                012310  001011                          BNE T24L04              ;TEST NEXT ADDRESS IF NO
                012312  032777  000003  170206          BIT #3,@BECR2           ;HAVE ADDRESS BITS 16,17 BEEN EXERCISED?
                012320  001422                          BEQ T24L03              ;GO RESTORE TRAPS IF YES
                012322  005277  170200                  INC @BECR2              ;INC ADDRESS BITS 16,17
                012326  042777  000004  170172          BIC #4,@BECR2           ;CLEAR BIT 2 OF BECR2 IF SET
                012334  012777  002041  170162  T24L04: MOV #2041,@BECR1        ;DO DATI NPR TO ADDRESS
                012342  004767  003676                  JSR PC,CRDY             ;WAIT TILL RDY SET
                012346  032777  001000  170152          BIT #1000,@BECR2        ;WAS WRONG ADDRESS LINES ERROR BIT SET?
                012354  001744                          BEQ T24L05              ;TEST NEXT ADDRESS IF NO
                012356  104070                          ERROR  D56              ;ERROR: TEST OF WRONG A LINES ERROR BIT FAILED
                012360  104071                          ERROR  D57              ;BECR2 BIT 9 FALSELY SET
                012362  004767  004042                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
                012366  004767  003630          T24L03: JSR PC,RCATCH           ;RESTORE TRAP CATCHER
```

```
                                ;;*****************************************************************
                                ;*TEST 31       TEST WRONG GRANT  AND NO GRANT OR NOT ONE GRANT ERROR BITS DO NOT SET
                                ;*
                                ;*THE UBE IS SET UP TO DO ONE DATI XFER/REQUEST.  ALL
                                ;*THE POSSIBLE COMBINATIONS OF BR AND NPR LEVELS ARE THEN
                                ;*EXERCISED.  AFTER EACH, THE ERROR BITS AND INTERRUPTS ARE
                                ;*CHECKED FOR.  FINALLY, A DATI NPR IS DONE FROM A BUFFER
                                ;*AREA WITH THE INTERRUPT ON DONE BIT SET.  UPON INTERRUPT, THE
                                ;*ERROR BITS ARE CHECKED.
                                ;;*****************************************************************
012372  000004                  TST31:  SCOPE
012374  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
012400  004767  003564                  JSR PC,CLRREG           ;CLEAR UBE REG
012404  012777  002000  170112          MOV #2000,@BECR1        ;SET UP UBE TO DO 1 DATI XFER/REQ.
012412  012777  012512  170112          MOV #T25L01,@INTVEC     ;SET UP FOR INTERRUPTS
012420  012737  000340  177776  T25L05: MOV     #340,   @#PSW           ;LOCK OUT INTERRUPTS
012426  012777  177777  170064          MOV #177777,@BECC       ;SET UBE TO DO 1 CYCLE
012434  012777  030010  170060          MOV #BUFF1,@BEBA        ;SET UBE TO ADDRESS BUFFER AREA
012442  062777  000003  170054          ADD #3,@BECR1           ;HAVE UBE DO NEXT LEVEL OF REQUEST
012450  005037  177776                  CLR     @#PSW           ;ALLOW DATA XFERS VIA BR AND NPR LEVELS
012454  004767  003564                  JSR PC,CRDY             ;WAIT TILL RDY SET
012460  032777  000076  170036          BIT #76,@BECR1          ;HAVE ALL REQUEST LEVELS BEEN EXERCISED
012466  001425                        BEQ T25L02              ;BRANCH IF YES
012470  032777  000040  170030          BIT #40,@BECR2          ;WAS WRONG GRANT ERROR BIT SET?
012476  001062                        BNE T25L03              ;BRANCH TO ERROR IF SET
012500  032777  002000  170020          BIT #2000,@BECR2        ;WAS NO GRANT OR NOT ONE GRANT ERROR BIT SET?
012506  001066                        BNE T25L04              ;BRANCH TO ERROR IF YES
012510  000743                        BR T25L05               ;GO TEST NEXT LEVEL
012512  104100                  T25L01: ERROR  D64              ;ERROR: TEST OF WRONG GRANT OR NOT ONE GRANT FAILED
012514  017767  170006  166472        MOV @BECR2,$REG0        ;SAVE ERROR BITS
012522  104077                        ERROR  D63              ;FALSELY INTERRUPTED CPU
012524  017767  167774  166462        MOV @BECR1,$REG0        ;SAVE BECR1
012532  104104                        ERROR  D68              ;WITH BECR1=
012534  004767  003670                JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
012540  000460                        BR T25L08               ;GO RESTORE TRAPS
012542  012777  012560  167762  T25L02: MOV #T25L06,@INTVEC   ;SET UP NEW INT. AREA
012550  012777  002143  167746        MOV #2143,@BECR1        ;HAVE UBE DO 1 DATI NPR AND INT ON DONE
012556  000001                        WAIT                    ;WAIT TO BE INTERRUPTED
012560  032777  000040  167740  T25L06: BIT #40,@BECR2         ;WAS WRONG GRANT ERROR BIT SET?
012566  001015                        BNE T25L07              ;BRANCH TO ERROR IF WAS
012570  032777  002000  167730        BIT #2000,@BECR2        ;WAS NO GRANT OR NOT ONE GRANT BIT SET?
012576  001441                        BEQ T25L08              ;GO RESTORE TRAPS IF WAS NOT
012600  104100                        ERROR  D64              ;ERROR: TEST OF WRONG GRANT OR NOT ONE GRANT FAILED
012602  017767  167716  166404        MOV @BECR1,$REG0        ;SAVE BECR1
012610  104101                        ERROR  D65              ;NO GRANT OR NOT ONE GRANT ERROR BIT FALSELY SET
012612  104102                        ERROR  D66              ;WITH INT ON DONE = 1
012614  004767  003610                JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
012620  000430                        BR T25L08               ;GO RESTORE TRAPS
012622  104100                  T25L07: ERROR  D64              ;ERROR: TEST OF WRONG GRANT OR NOT ONE GRANT FAILED
012624  017767  167674  166362        MOV @BECR1,$REG0        ;SAVE BECR1
012632  104103                        ERROR  D67              ;WRONG GRANT ERROR BIT FALSELY SET
012634  104102                        ERROR  D66              ;WITH INT ON DONE = 1
012636  004767  003566                JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
012642  000417                        BR T25L08               ;GO RESTORE TRAPS
012644  104100                  T25L03: ERROR  D64              ;ERROR: TEST OF WRONG GRANT OR NOT ONE GRANT FAILED
012646  017767  167652  166340        MOV @BECR1,$REG0        ;SAVE BECR1
```

```
2056  012654  104103                          ERROR  D67              ;WRONG GRANT ERROR BIT FALSELY SET
2057  012656  004767  003546                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2058  012662  000407                          BR T25L08               ;GO RESTORE TRAPS
2059  012664  104100                  T25L04: ERROR  D64              ;ERROR: TEST OF WRONG GRANT OR NOT ONE GRANT FAILED
2060  012666  017767  167632  166320          MOV @BECR1,$REG0        ;SAVE BECR1
2061  012674  104101                          ERROR  D65              ;NO GRANT OR NOT ONE GRANT ERROR BIT FALSELY SET
2062  012676  004767  003526                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2063  012702  004767  003314          T25L08: JSR PC,RCATCH           ;RESTORE TRAP CATCHER
2064
2065                                  ;;**********************************************************************
2066                                  ;*TEST 32       TEST TIME DELAY AND BUSS LATENCY ERROR BITS
2067                                  ;*
2068                                  ;*THE BUS LATENCY ERROR BIT IS SET BY DOING A RELEASE
2069                                  ;*BUS IMMEDIATE FUNCTION AND SETTING THE TIME DELAY BIT.  THE
2070                                  ;*ERROR BIT AND BIT 15 OF BECR1 ARE CHECKED TO SET.  THE
2071                                  ;*ERROR INTERRUPT IS THEN CHECKED FOR AND THE ERROR CONDITION
2072                                  ;*IS TESTED TO CLEAR.
2073                                  ;;**********************************************************************
2074  012706  000004                  TST32:  SCOPE
2075  012710  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
2076  012714  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
2077  012722  004767  003242                  JSR PC,CLRREG           ;CLEAR UBE REG
2078  012726  012777  040000  167572          MOV #40000,@BECR2       ;SET TIME DELAY BIT
2079  012734  012777  013042  167570          MOV #T26L01,@INTVEC     ;SET UP FOR INTERRUPTS
2080  012742  012777  006003  167554          MOV #6003,@BECR1        ;DO RELEASE BUS IMMED.
2081  012750  005000                          CLR R0                  ;INITIALIZE R0
2082  012752  005200                  T26L02: INC R0                  ;DELAY TO WAIT FOR
2083  012754  022700  000400                  CMP #400,R0             ;BUSS LATENCY ERROR BIT
2084  012760  001374                          BNE T26L02              ;TO SET
2085  012762  032777  000100  167536          BIT #100,@BECR2         ;WAS BUSS LATENCY ERROR BIT SET?
2086  012770  C01004                          BNE T26L03              ;BRANCH IF YES
2087  012772  104106                          ERROR  D70              ;ERROR: TEST OF TIME DALAY AND BUSS LATENCY FAILED
2088  012774  104107                          ERROR  D71              ;TO SET BIT 6 OF BECR2
2089  012776  004767  003426                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2090  013002  032777  100000  167514  T26L03: BIT #100000,@BECR1      ;WAS ERROR BIT SET?
2091  013010  001004                          BNE T26L04              ;BRANCH IF YES
2092  013012  104106                          ERROR  D70              ;ERROR: TEST OF TIME DELAY AND BUSS LATENCY FAILED
2093  013014  104075                          ERROR  D61              ;TO SET BIT 15 OF BECR1
2094  013016  004767  003406                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2095  013022  005037  177776          T26L04: CLR @#PSW               ;ALLOW ERROR INTERRUPTS
2096  013026  000240                          NOP                     ;UBE SHOULD INTERRUPT
2097  013030  104106                          ERROR  D70              ;ERROR: TEST OF TIME DELAY AND BUSS LATENCY FAILED
2098  013032  104072                          ERROR  D58              ;TO INTERRUPT CPU
2099  013034  004767  003370                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2100  013040  000412                          BR T26L05               ;GO TO END OF TEST
2101  013042  005077  167462          T26L01: CLR @BERE               ;CLEAR ERROR CONDITION
2102  013046  032777  000100  167452          BIT #100,@BECR2         ;WAS ERROR CLEARED?
2103  013054  001404                          BEQ T26L05              ;BRANCH IF YES
2104  013056  104106                          ERROR  D70              ;ERROR: TEST OF TIME DELAY AND BUSS LATENCY FAILED
2105  013060  104110                          ERROR  D72              ;TO CLEAR BIT 6 OF BECR2
2106  013062  004767  003342                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2107  013066  004767  003076          T26L05: JSR PC,CLRREG           ;CLEAR ALL UBE REG
2108  013072  004767  003176                  JSR PC,DINT             ;DISREGARD ERROR INTERRUPTS
2109  013076  012777  177777  167414          MOV #177777,@BECC       ;HAVE UBE DO DATI
2110  013104  012777  030010  167410          MOV #BUFF1,@BEBA        ;SO BUSS LATENCY REG
2111  013112  012777  002041  167404          MOV #2041,@BECR1        ;HOLD FLOP CLEARED
```

```
2112  013120  004767  003120              JSR PC,CRDY             ;WAIT FOR RDY SET
2113  013124  005077  167400              CLR @BERE               ;CLEAR LATENCY ERROR IF SET
2114  013130  004767  003066              JSR PC,RCATCH           ;RESTORE TRAPS
2115
2116                                      ;;**********************************************************************
2117                                      ;*TEST 33       TEST MULTIPLE INTERRUPTS SET RDY BIT
2118                                      ;;**********************************************************************
2119  013134  000004                      TST33:  SCOPE
2120  013136  012706  001100              MOV #STACK,SP           ;INITIALIZE STACK
2121  013142  004767  003022              JSR  PC,CLRREG          ;CLEAR ALL UBE REG
2122  013146  004767  003122              JSR  PC,DINT            ;DISREGARD INTERRUPTS
2123  013152  005037  177776              CLR @#PSW               ;ALLOW INTERRUPTS
2124  013156  012777  177776  167334      MOV #177776,@BECC       ;HAVE UBE DO 2 CYCLES
2125  013164  012777  040000  167334      MOV #40000,@BECR2       ;DO TIME DLY
2126  013172  012777  000003  167324      MOV #3,@BECR1           ;HAVE UBE INT. VIA BR4
2127  013200  004767  003040              JSR PC,CRDY             ;CHECK FOR RDY SET
2128  013204  005704                      TST R4                  ;WAS RDY SET?
2129  013206  001403                      BEQ T31L01              ;BRANCH IF YES
2130  013210  104124                      ERROR   D84             ;ERROR:TEST OF MULTIPLE INTERRUPTS FAILED TO SET RDY
2131  013212  004767  003212              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2132  013216  004767  003000              T31L01: JSR PC,RCATCH           ;RESTORE TRAP CATCHER
2133
2134                                      ;;**********************************************************************
2135                                      ;*TEST 34       TEST POWER DOWN SEQUENCE
2136                                      ;*
2137                                      ;*THE POWER DOWN TEST IS ONLY DONE IF SW4=1.
2138                                      ;*THE POWER DOWN IS TESTED FOR AND THEN THE POWER UP
2139                                      ;*IS TESTED.  AN INTERNAL REG R0 COUNTS FOR A TIME >150
2140                                      ;*MS TO SEE IF THE CPU GETS POWERED UP.  THE PROGRAM
2141                                      ;*THEN WAITS FOR A TIME >150MS TO SEE IF THE CPU
2142                                      ;*GETS POWERED DOWN AGAIN.
2143                                      ;;**********************************************************************
2144  013222  000004                      TST34:  SCOPE
2145  013224  012767  000001  166002              MOV     #1,$TIMES       ;;DO 1 ITERATION
2146  013232  032777  000020  165732              BIT #20,@SWR            ;SEE IF POWER DOWN TO BE TESTED
2147  013240  001516                              BEQ     TST35           ;;GO TO NEXT TEST IF SWR4 = 0
2148  013242  012737  000340  177776              MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
2149  013250  012706  001100                      MOV #STACK,SP           ;INITIALIZE STACK
2150  013254  013746  000024                      MOV @#24,-(SP)          ;SAVE POWER FAIL VECTOR ON STACK
2151  013260  013746  000026                      MOV @#26,-(SP)          ;SAVE POWER FAIL VECTOR ON STACK
2152  013264  012737  013322  000024              MOV #T27L01,@#24        ;SET UP FOR POWER FAIL
2153  013272  012737  000340  000026              MOV #340,@#26           ;SET UP FOR POWER FAIL
2154  013300  012777  000020  167220              MOV #20,@BECR2          ;HAVE UBE DO POWER FAIL
2155  013306  000240                              NOP                     ;SHOULD POWER FAIL HERE
2156  013310  104111                              ERROR  D73              ;ERROR: TEST OF POWER DOWN BIT FAILED
2157  013312  104112                              ERROR  D74              ;TO POWER DOWN CPU
2158  013314  004767  003110                      JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2159  013320  000450                              BR T27L02               ;RESTORE TRAPS
2160  013322  022626                      T27L01: CMP (SP)+,(SP)+         ;RESTORE STACK
2161  013324  012737  013366  000024              MOV #T27L03,@#24        ;SET UP FOR POWER UP SEQUENCE
2162  013332  005000                              CLR R0                  ;INITIALIZE COUNTER
2163  013334  005001                              CLR R1                  ;INITIALIZE COUNTER
2164  013336  005200                      T27L04: INC R0                  ;COUNT FOR A TIME
2165  013340  005700                              TST R0                  ;GREATER THAN 150 MS
2166  013342  001375                              BNE T27L04
2167  013344  005201                              INC R1
```

```
2168  013346  022701  000004              CMP #4,R1               ;IS TIME > 150 MS?
2169  013352  001371                      BNE T27L04              ;BRANCH IF NO
2170  013354  104111                      ERROR  D73              ;ERROR: TEST OF POWER DOWN BIT FAILED
2171  013356  104113                      ERROR  D75              ;TO POWER UP CPU
2172  013360  004767  003044              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2173  013364  000426                      BR T27L02               ;RESTORE TRAPS
2174  013366  012737  013430  000024  T27L03: MOV #T27L05,@#24    ;SET UP TO POWER DOWN AGAIN
2175  013374  005000                      CLR R0
2176  013376  005001                      CLR R1
2177  013400  005200              T27L06: INC R0                  ;COUNT FOR A TIME
2178  013402  005700                      TST R0                  ;GREATER THAN 150 MS
2179  013404  001375                      BNE T27L06
2180  013406  005201                      INC R1
2181  013410  022701  000004              CMP #4,R1               ;IS TIME > 150 MS?
2182  013414  001371                      BNE T27L06              ;BRANCH IF NO
2183  013416  104111                      ERROR  D73              ;ERROR: TEST OF POWER DOWN BIT FAILED
2184  013420  104114                      ERROR  D76              ;TO REPOWER DOWN CPU
2185  013422  004767  003002              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2186  013426  000405                      BR T27L02               ;GO CHECK POWER DOWN BIT
2187  013430  022626              T27L05: CMP (SP)+,(SP)+         ;RESTORE STACK
2188  013432  012737  013442  000024      MOV #T27L02,@#24        ;SET UP TO POWER UP AGAIN
2189  013440  000001                      WAIT                    ;WAIT TO POWER UP AGAIN
2190  013442  032777  000020  167056  T27L02: BIT #20,@BECR2      ;WAS POWER DOWN BIT SET?
2191  013450  001004                      BNE T27L07              ;BRANCH IF YES
2192  013452  104111                      ERROR  D73              ;ERROR: TEST OF POWER DOWN BIT FAILED
2193  013454  104115                      ERROR  D77              ;TO SET BIT 4 OF BECR2
2194  013456  004767  002746              JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2195  013462  012637  000026  T27L07: MOV (SP)+,@#26             ;RESTORE POWER FAIL VECTOR
2196  013466  012637  000024              MOV (SP)+,@#24
2197  013472  005077  167030              CLR @BECR2              ;CLEAR POWER DOWN BIT
2198
2199                                  ;;*********************************************************************
2200                                  ;*TEST 35       TEST DCLO CLEARS BECC, BEBA, BECR2 AND BITS 0-6, 7-15 OF BECR1
2201                                  ;*
2202                                  ;*THIS TEST IS ONLY DONE IF SW4=1.
2203                                  ;;*********************************************************************
2204  013476  000004              TST35:  SCOPE
2205  013500  012767  000001  165526      MOV     #1,$TIMES       ;;DO 1 ITERATION
2206  013506  032777  000020  165456      BIT #20,@SWR            ;SEE IF POWER DOWN TO BE TESTED
2207  013514  001002                      BNE T28L10              ;BRANCH IF SW4=1
2208  013516  000167  000410              JMP TSTB                ;GO TO NEXT TEST
2209  013522  012777  177777  166770  T28L10: MOV #177777,@BECC  ;HAVE UBE DO 1 CYCLE
2210  013530  012777  000003  166770      MOV #3,@BECR2           ;SET ADDRESS BITS 16, 17
2211  013536  012777  160000  166756      MOV #160000,@BEBA       ;LOAD UBE WITH ADDRESS THAT RETURNS NO SSYN
2212  013544  004767  002524              JSR PC,DINT             ;DISREGARD INTERRUPTS
2213  013550  012777  002041  166746      MOV #2041,@BECR1        ;HAVE UBE DO DATI SO CCOVF=1 AND NSSYN ERROR = 1
2214  013556  005037  177776              CLR @#PSW               ;ALLOW INTERRUPTS
2215  013562  000001                      WAIT                    ;WAIT TILL ERROR INTERRUPT
2216  013564  013746  000024              MOV @#24,-(SP)          ;STORE POWER VECTOR ON STACK
2217  013570  013746  000026              MOV @#26,-(SP)          ;STORE POWER VECTOR ON STACK
2218  013574  012777  177777  166720      MOV #177777,@BEBA       ;LOAD ADDRESS REG WITH ALL "1"
2219  013602  012777  177777  166710      MOV #177777,@BECC       ;LOAD CYCLE COUNT REG WITH ALL "1"
2220  013610  012777  077776  166706      MOV #77776,@BECR1       ;LOAD BECR1 WITH ONES
2221  013616  012737  013634  000024      MOV #T28L01,@#24        ;SET UP FOR POWER DOWN
2222  013624  012777  040037  166674      MOV #40037,@BECR2       ;LOAD BECR2 WITH ONES AND DO POWER DOWN
2223  013632  000001                      WAIT                    ;CPU SHOULD POWER DOWN
```

```
2224  013634  022626                    T28L01: CMP (SP)+,(SP)+         ;RESTORE STACK
2225  013636  012737  013646  000024            MOV #T28L05,@#24        ;SETUP FOR POWER UP
2226  013644  000001                            WAIT                    ;CPU SHOULD POWER UP
2227  013646  042777  000020  166652  T28L05: BIC #20,@BECR2            ;CLEAR POWER DOWN BIT
2228  013654  016767  166640  165332            MOV BECC,$REG0          ;SAVE BECC ADDRESS
2229  013662  005067  165332                    CLR $REG2               ;SAVE CORRECT DATA
2230  013666  005777  166626                    TST @BECC               ;(BECC)=0?
2231  013672  001026                            BNE T28L02              ;BRANCH IF NO
2232  013674  016767  166622  165312            MOV BEBA,$REG0          ;SAVE BEBA ADDRESS
2233  013702  005777  166614                    TST @BEBA               ;(BEBA)=0?
2234  013706  001020                            BNE T28L02              ;BRANCH IF NO
2235  013710  016767  166612  165276            MOV BECR2,$REG0         ;SAVE BECR2 ADDRESS
2236  013716  005777  166604                    TST @BECR2              ;WAS BECR2 CLEARED?
2237  013722  001012                            BNE T28L02              ;BRANCH IF NO
2238  013724  016767  166574  165262            MOV BECR1,$REG0         ;SAVE BECR1 ADDRESS
2239  013732  012767  000200  165260            MOV #200,$REG2          ;SAVE CORRECT DATA (BECR1)
2240  013740  022777  000200  166556            CMP #200,@BECR1         ;WAS BECR1 CLEARED?
2241  013746  001407                            BEQ T28L03              ;BRANCH IF YES
2242  013750  017767  165240  165240  T28L02: MOV @$REG0,$REG1         ;SAVE BAD DATA
2243  013756  104116                            ERROR  D78              ;ERROR: DCLO FAILED TO CLEAR REG
2244  013760  004767  002444                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2245  013764  000454                            BR T28L04               ;GO RESTORE VECTORS
2246  013766  012737  000340  177776  T28L03: MOV #340,@#PSW           ;LOCK OUT INTERRUPTS
2247  013774  012777  040000  1[illegible]4524            MOV #40000,@BECR2       ;SET TIME DLY BIT
2248  014002  012777  006003  1[illegible] 514            MOV #6003,@BECR1        ;DO RELEASE BUSS IMMED. TO SET LATENCY ERROR BIT
2249  014010  032777  000100  166510  T28L06: BIT #100,@BECR2           ;TEST LATENCY ERROR BIT
2250  014016  001774                            BEQ T28L06              ;WAIT TILL IT SETS
2251  014020  005037  177776                    CLR @#PSW               ;ALLOW LATENCY ERROR INTERRUPT
2252  014024  000240                            NOP                     ;ALLOW INTERRUPT TO BE IGNORED
2253  014026  012737  014044  000024            MOV #T28L08,@#24        ;SET UP FOR POWER DOWN
2254  014034  052777  000020  166464            BIS #20,@BECR2          ;SET POWER DOWN BIT
2255  014042  000001                            WAIT                    ;WAIT FOR POWER DOWN
2256  014044  022626                    T28L08: CMP (SP)+,(SP)+         ;RESTORE STACK
2257  014046  012737  014056  000024            MOV #T28L09,@#24        ;SETUP FOR POWER UP
2258  014054  000001                            WAIT                    ;CPU SHOULD POWER UP
2259  014056  005077  166444  T28L09: CLR @BECR2                        ;CLEAR POWER DOWN BIT
2260  014062  005777  166440                    TST @BECR2              ;WAS BUSS LATENCY ERROR BIT CLEARED?
2261  014066  001413                            BEQ T28L04              ;BRANCH IF YES
2262  014070  016767  166432  165116            MOV BECR2,$REG0         ;SAVE REG ADDRESS
2263  014076  017767  166424  165112            MOV @BECR2,$REG1        ;SAVE REG DATA
2264  014104  005067  165110                    CLR $REG2               ;SAVE CORRECT DATA
2265  014110  104116                            ERROR  D78              ;ERROR: DCLO FAILED TO CLEAR REG
2266  014112  004767  002312                    JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2267  014116  004767  002100            T28L04: JSR PC,RCATCH           ;RESTORE TRAP CATCHER
2268  014122  012637  000026                    MOV (SP)+,@#26          ;RESTORE POWER VECTOR
2269  014126  012637  000024                    MOV (SP)+,@#24          ;RESTORE POWER VECTOR
2270
2271  014132                            TSTB:
2272
2273                                    ;;**********************************************************************
2274                                    ;*TEST 36       TEST SIMULTANEOUS GO ADDRESS
2275                                    ;*
2276                                    ;*THE UBE IS SETUP TO INTERRUPT ON LEVEL 7 AND
2277                                    ;*THEN TOLD TO GO VIA THE SIMULTANEOUS GO.  NO
2278                                    ;*INTERRUPT INDICATES AN ERROR.
2279                                    ;;**********************************************************************
```

```
 2280  014132  000004                  TST36:  SCOPE
 2281  014134  012706  001100                  MOV #STACK,SP           ;RESTORE STACK
 2282  014140  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INTERRUPTS
 2283  014146  004767  002016                  JSR PC,CLRREG           ;CLEAR ALL UBE REGS.
 2284  014152  012777  014210  166352          MOV #T09L01,@INTVEC     ;SETUP TO RECEIVE INTERRUPT
 2285  014160  012777  000020  166336          MOV #20,@BECR1          ;SETUP TO DO BR=7
 2286  014166  005277  166342                  INC @BEGO               ;START SIMULTANEOUS GO
 2287  014172  012737  000300  177776          MOV #300,@#PSW          ;ALLOW INTERRUPTS
 2288  014200  000240                          NOP                     ;UBE SHOULD INTERRUPT HERE
 2289  014202  104025                          ERROR  D21              ;ERROR: SIMULTANEOUS GO FAILED
 2290  014204  004767  002220                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 2291  014210  004767  002006          T09L01: JSR PC,RCATCH           ;RESTORE TRAP CATCHER
 2292
 2293
```

```
 2294                                   ;;*****************************************************************
 2295                                   ;*TEST 37       DYNAMIC TEST OF UBE
 2296                                   ;*
 2297                                   ;*THIS TEST EXERCISES THE MOST HARDWARE IN THE
 2298                                   ;*UBE AT ONE TIME.  THE EXERCISOR IS SET UP TO DO EIGHT
 2299                                   ;*DATOB ON DATIP XFERS VIA NPR AND INTERRUPT ON DONE.
 2300                                   ;*AFTER INTERRUPTING, A BUFFER AREA IS EXAMINED TO SEE IF
 2301                                   ;*THE OPERATIONS WERE DONE PROPERLY.  THE ABOVE IS THEN
 2302
 2303                                   ;*REPEATED 100 TIMES.
 2304
 2305                                   ;;*****************************************************************
 2306  014214  000004                  TST37:  SCOPE
 2307  014216  012767  000001  165010          MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
2308  014224  004767  001740            JSR PC,CLRREG           ;CLEAR UBE REG
2309  014230  005002                    CLR R2                  ;INITIALIZE COUNT
2310  014232  005037  177776            CLR @#PSW               ;ALLOW INTERRUPTS
2311  014236  012700  030010  T29L04:   MOV #BUFF1,R0           ;GET BUFFER ADDRESS
2312  014242  012720  052525  T29L01:   MOV #52525,(R0)+        ;LOAD BUFFER
2313  014246  020027  030032            CMP R0,#BUFF1+22        ;ENTIRE BUFFER LOADED?
2314  014252  001373                    BNE T29L01              ;BRANCH IF NO
2315  014254  012777  014334  166250    MOV #T29L02,@INTVEC     ;SET UP FOR INTERRUPTS
2316  014262  012777  030010  166232    MOV #BUFF1,@BEBA        ;LOAD BUFF ADDRESS IN UBE
2317  014270  012777  177760  166222    MOV #177760,@BECC       ;SET UBE TO DO 16 CYCLES
2318  014276  012777  042561  166220    MOV #42561,@BECR1       ;DO DATOB ON DATIP, AND INT. VIA BR7 WHEN DONE
2319  014304  005000                    CLR R0                  ;INITIALIZE COUNTER
2320  014306  016767  013516  013514  T29L06:  MOV BUFF1+20,BUFF1+20  ;DO BACKGROUND NOISE PATTERN
2321  014314  005200                    INC R0                  ;WAIT FOR COUNTER R0
2322  014316  005700                    TST R0                  ;TO OVERFLOW. IF DOES
2323  014320  001372                    BNE T29L06              ;UBE FAILED TO INTERRUPT
2324  014322  104120                    ERROR  D80              ;ERROR: DYNAMIC TEST OF UBE FAILED
2325  014324  104072                    ERROR  D58              ;TO INTERRUPT CPU
2326  014326  004767  002076            JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2327  014332  000432                    BR T29L07               ;GO RESTORE TRAPS CATCHER
2328  014334  022626  T29L02:   CMP (SP)+,(SP)+                 ;RESTORE STACK
2329  014336  012700  030010            MOV #BUFF1,R0           ;GET BUFFER ADDRESS
2330  014342  022710  125652  T29L05:   CMP #125652,(R0)        ;WAS DATA SHIFTED PROPERLY?
2331  014346  001011                    BNE T29L03              ;BRANCH TO ERROR IF NO
2332  014350  005720                    TST (R0)+               ;INC R0 BY 2
2333  014352  022700  030030            CMP #BUFF1+20,R0        ;AT END OF BUFFER?
2334  014356  001371                    BNE T29L05              ;BRANCH IF NO
2335  014360  005202                    INC R2                  ;UPDATE COUNT
2336  014362  020227  000100            CMP R2,#100             ;WAS UBE EXERCISED 100 TIMES?
2337  014366  001323                    BNE T29L04              ;BRANCH IF NO
2338  014370  000413                    BR      T29L07          ;RESTORE TRAPS
2339  014372  010067  164616  T29L03:   MOV R0,$REG0            ;SAVE ADDRESS
2340  014376  011067  164614            MOV (R0),$REG1          ;SAVE BAD DATA
2341  014402  012767  125652  164610    MOV #125652,$REG2       ;SAVE CORRECT DATA
2342  014410  104120                    ERROR  D80              ;ERROR: DYNAMIC TEST OF UBE FAILED
2343  014412  104121                    ERROR  D81              ;TO LOAD PROPER DATA
2344  014414  004767  002010            JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
2345  014420  004767  001576  T29L07:   JSR PC,RCATCH           ;RESTORE TRAP CATCHER
2346
2347                                    ;////////////////////////////////////////////////////////////////////////
2348                                    ;RETURN ROUTINE TO TEST NEXT UBE BEFORE DO LAST TEST
2349                                    ;////////////////////////////////////////////////////////////////////////
2350  014424  000004                    SCOPE                   ;SCOPE FOR PREVIOUS TEST
2351  014426  004767  001536  NUBE:    JSR PC,CLRREG           ;CLEAR UBE SO NO INT.
2352  014432  000167  166740  NUBE1:   JMP ACALC               ;GO SEE IF MORE UBE
2353
2354  014436  012767  014460  164474  LAST:   MOV #LAST1,$LPADR      ;SETUP LOOP ADDRESS FOR LAST TEST
2355  014444  012767  014460  164470          MOV #LAST1,$LPERR      ;SETUP LOOP ON ERROR ADDRESS FOR LAST TEST
2356  014452  105367  164456                  DECB $TSTNM            ;ADJUST TEST NUMBER
2357
2358
2359
2360                                    ;;******************************************************************
2361                                    ;*TEST 40       TEST PASSING OF GRANTS
2362                                    ;*
2363                                    ;*THIS TEST IS ONLY RUN IF THERE ARE MORE THAN ONE
```

```
                                ;*UBE.IT IS COMPOSED OF TWO PARTS.THE FIRST PART CHECKS THAT
                                ;*A HIGHER ELECTRICAL PRIORITY UBE WITH ALL BR LEVELS =1
                                ;*AND GO BIT =0 WILL PASS A GRANT TO THE NEXT LOWER ONE.
                                ;*THEN THIS SAME UBE IS CHECKED TO ALSO PASS A GRANT WHEN ALL BR=0
                                ;*AND THE GO BIT IS ENABLED.
                                ;*      THE SECOND PART VERIFIES THAT A UBE WITH A HIGHER ELECTRICAL PRIORITY
                                ;*BUT DOING A LOWER BR THAN A UBE OF LOWER ELECTRICAL
                                ;*PRIORITY, WILL PASS THE GRANT TO THE UBE OF LOWER ELECTRICAL
                                ;*PRIORITY.
                                ;*
                                ;*NOTE:  THE UBE WITH THE LOWEST ELECTRICAL PRIORITY
                                ;*       ON THE BUS MUST BE SWAPPED WITH A HIGHER
                                ;*       ONE AND THEN THE ENTIRE PROGRAM RERUN INORDER
                                ;*       THAT ITS PASSING GRANT LOGIC IS TESTED.
                                ;;*****************************************************************
014456  000004                 TST40:  SCOPE
014460  005767  166070         LAST1:  TST BE2BD               ;IS THERE MORE THAN ONE EXERCISOR?
014464  001013                         BNE T30L01              ;BRANCH IF YES
014466  032777  010000  164476         BIT #SW12,@SWR          ;INHIBIT TYPEOUTS?
014474  001005                         BNE 1$                  ;BRANCH IF YES
014476  104401  027304                 TYPE ,MSG11             ;PASSING OF GRANTS NOT TESTED WITH 1 EXERCISOR
014502  012767  000001  164524         MOV #1,$TIMES           ;DO 1 ITERATION IF THIS TEST NOT DONE
014510  000167  001332         1$:     JMP $EOP                ;GO TO END OF TEST

                                ;DETERMINE ELECTRICAL PRIORITY OF EXERCISORS

014514  012706  001100         T30L01: MOV #STACK,SP           ;INITIALIZE STACK
014520  012777  014726  166024         MOV #T30L02,@BE1VEC     ;SET UP UBE1 INTERRUPT HANDLER
014526  016700  166020                 MOV BE1VEC,R0
014532  012760  000340  000002         MOV #340,2(R0)
014540  012777  014742  166022         MOV #T30L03,@BE2VEC     ;SET UP UBE2 INTERRUPT HANDLER
014546  016700  166016                 MOV BE2VEC,R0
014552  012760  000340  000002         MOV #340,2(R0)
014560  005767  166022                 TST BE3VEC              ;ARE THERE 3 UBE?
014564  001423                         BEQ T30L21              ;BRANCH IF NO
014566  012777  014756  166012         MOV #T30L04,@BE3VEC     ;SET UP UBE3 INTERRUPT HANDLER
014574  016700  166006                 MOV BE3VEC,R0
014600  012760  000340  000002         MOV #340,2(R0)
014606  005767  166012                 TST BE4VEC              ;ARE THERE 4 UBE?
014612  001410                         BEQ T30L21              ;BRANCH IF NO
014614  012777  014772  166002         MOV #T30L05,@BE4VEC     ;SET UP UBE4 INTERRUPT HANDLER
014622  016700  165776                 MOV BE4VEC,R0
014626  012760  000340  000002         MOV #340,2(R0)
014634  012700  030010         T30L21: MOV #BUFF1,R0           ;GET BUFFER ADDRESS
014640  005001                         CLR R1                  ;INITIALIZE COUNT OF INTERRUPTS
014642  012737  000340  177776         MOV #340,@#PSW          ;SET PSW PRIORITY=7
014650  012777  000020  165666         MOV #20,@BE1CR1         ;LOAD FIRST UBE TO DO INT. VIA BR7
014656  012777  000020  165676         MOV #20,@BE2CR1         ;LOAD SECOND UBE TO DO INT. VIA BR7
014664  005767  165710                 TST BE3CR1              ;TEST IF 3 EXERCISORS
014670  001411                         BEQ T30L07              ;BRANCH IF NO
014672  012777  000020  165700         MOV #20,@BE3CR1         ;LOAD THIRD UBE TO DO INT. VIA BR7
014700  005767  165712                 TST BE4CR1              ;TEST IF 4 EXERCISORS
014704  001403                         BEQ T30L07              ;BRANCH IF NO
014706  012777  000020  165702         MOV #20,@BE4CR1         ;LOAD FOURTH UBE TO DO INT. VIA BR7
014714  005277  165614         T30L07: INC @BEGO               ;LET ALL EXERCISORS INTERRUPT
014720  005037  177776                 CLR @#PSW               ;ALLOW INTERRUPTS
```

```
2420  014724  000001                          WAIT                    ;WAIT FOR 1ST INTERRUPT
2421  014726  012720  002536          T30L02: MOV #BE1BD,(R0)+        ;LOAD BUFFER WITH POINTER TO ADDRESS OF UBE
2422  014732  012777  006002  165604          MOV #6002,@BE1CR1       ;SETUP FIRST UBE TO DO A FUN 3
2423  014740  000421                          BR T30L06               ;GO SEE IF ALL UBE INTERRUPTED
2424  014742  012720  002554          T30L03: MOV #BE2BD,(R0)+        ;LOAD BUFFER WITH POINTER TO UBE ADDRESSES
2425  014746  012777  006002  165606          MOV #6002,@BE2CR1       ;SETUP SECOND UBE TO DO A FUN3
2426  014754  000413                          BR T30L06               ;GO SEE IF ALL UBE INTERRUPED
2427  014756  012720  002572          T30L04: MOV #BE3BD,(R0)+        ;LOAD BUFFER WITH POINTER TO UBE ADDRESS
2428  014762  012777  006002  165610          MOV #6002,@BE3CR1       ;SETUP THIRD UBE TO DO A FUN3
2429  014770  000405                          BR T30L06               ;GO SEE IF ALL UBE INTERRUPTED
2430  014772  012720  002610          T30L05: MOV #BE4BD,(R0)+        ;LOAD BUFFER WITH POINTER TO UBE ADDRESS
2431  014776  012777  006002  165612          MOV #6002,@BE4CR1       ;SETUP FOURTH UBE TO DO A FUN3
2432  015004  022626                  T30L06: CMP (SP)+,(SP)+         ;RESTORE STACK
2433  015006  005201                          INC R1                  ;COUNT INTERRUPTS
2434  015010  020167  165612                  CMP R1,UCNT             ;HAVE ALL EXERCISORS INTERRUPTED?
2435  015014  001403                          BEQ T30L22              ;BRANCH IF YES
2436  015016  005037  177776                  CLR @#PSW               ;ALLOW NEXT UBE TO INTERRUPT
2437  015022  000001                          WAIT                    ;WAIT FOR INTERRUPT
2438  015024  024040                  T30L22: CMP -(R0),-(R0)         ;DECREMENT R0 BY 4
2439  015026  011067  012766                  MOV (R0),BUFF1+10       ;PUT NEXT TO LOWEST PRIORITY POINTER IN BUFF1+10
2440
2441                                  ;BUFFER NOW CONTAINS VECTORS IN ORDER OF ELECTRICAL PRIORITY
2442
2443                                  ;PART 1
2444
2445  015032  016700  165570                  MOV UCNT,R0             ;GET COUNT OF UBE
2446  015036  005300                          DEC R0                  ;ADJUST COUNT
2447  015040  005001                          CLR R1                  ;CLEAR INDEX REG
2448  015042  016102  030010          T30L28: MOV BUFF1(R1),R2        ;GET PTER TO ADDRESS OF HIGHER PRIORITY UBE
2449  015046  012772  000036  000006          MOV #36,@6(R2)          ;SET ALL BR =1 IN THIS UBE
2450  015054  005721                          TST (R1)+               ;UPDATE INDEX
2451  015056  016103  030010                  MOV BUFF1(R1),R3        ;GET PTER TO ADDRESS OF NEXT LOWER PRIORITY UBE
2452  015062  012773  015200  000014          MOV #T30L25,@14(R3)     ;SET UP FOR INT.
2453  015070  012773  000002  000006  T30L30: MOV #2,@6(R3)           ;SETUP LOWER PRIORITY UBE FOR BR4
2454  015076  005273  000006          T30L26: INC @6(R3)              ;HAVE UBE INT.
2455  015102  005037  177776                  CLR @#PSW               ;ALLOW INT.
2456  015106  000240                          NOP                     ;SHOULD INT. HERE
2457  015110  012737  000340  177776          MOV #340,@#PSW          ;LOCK OUT INT.
2458  015116  104122                  T30L29: ERROR  D82              ;ERROR:TEST OF PASSING GRANTS FAILED
2459  015120  032777  020000  164044          BIT #SW13,@SWR          ;INHIBIT ERROR TYPEOUTS?
2460  015126  001022                          BNE 1$                  ;BRANCH IF YES
2461  015130  016367  000014  164056          MOV 14(R3),$REG0        ;SAVE INT. VECTOR
2462  015136  104401  027156                  TYPE ,MSG7              ;UBE WITH INT. VECTOR:
2463  015142  016746  164046                  MOV     $REG0,-(SP)     ;;SAVE $REG0 FOR TYPEOUT
2464  015146  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2465  015150  017367  000006  164040          MOV @6(R3),$REG1        ;SAVE (BECR1)
2466  015156  104401  026135                  TYPE ,DH65              ;WITH BECR1=
2467  015162  016746  164030                  MOV     $REG1,-(SP)     ;;SAVE $REG1 FOR TYPEOUT
2468  015166  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2469  015170  104401  027251                  TYPE ,MSG10             ;SHOULD HAVE INT.
2470  015174  000167  000474          1$:     JMP T30L12              ;GO TO END OF TEST
2471  015200  006373  000006          T30L25: ASL @6(R3)              ;DO NEXT BR LEVEL
2472  015204  042773  000400  000006          BIC #400,@6(R3)         ;CLEAR SHIFTED RDY BIT
2473  015212  032773  000040  000006          BIT #40,@6(P3)          ;ALL BR TESTED?
2474  015220  001726                          BEQ T30L26              ;BRANCH IF NO
2475
```

```
 2476  015222  012773  015246  000014          MOV #T30L27,@14(R3)     ;SETUP FOR INT.
 2477  015230  012772  015116  000014          MOV #T30L29,@14(R2)     ;SETUP FOR ERROR INT.
 2478  015236  012772  000001  000006          MOV #1,@6(R2)           ;HAVE HIGHER UBE TRY TO INT.
 2479  015244  000711                          BR T30L30               ;LET LOWER UBE INT.
 2480
 2481  015246  006373  000006          T30L27: ASL @6(R3)              ;DO NEXT LEVEL BR
 2482  015252  042773  000400  000006          BIC #400,@6(R3)         ;CLEAR SHIFTED RDY
 2483  015260  032773  000040  000006          BIT #40,@6(R3)          ;ALL BR TESTED?
 2484  015266  001703                          BEQ T30L26              ;BRANCH IF NO
 2485  015270  012772  006003  000006          MOV #6003,@6(R2)        ;HAVE HIGHER UBE DO FUN3
 2486  015276  005037  177776                  CLR @#PSW               ;ALLOW REQUESTS
 2487  015302  105772  000006          1$:     TSTB @6(R2)             ;IS UBE DONE?
 2488  015306  100375                          BPL 1$                  ;BRANCH IF NO
 2489  015310  012737  000340  177776          MOV #340,@#PSW          ;SET LEVEL =7
 2490  015316  005300                          DEC R0                  ;ADJUST UBE COUNT
 2491  015320  005700                          TST R0                  ;ALL UBE TESTED?
 2492  015322  001247                          BNE T30L28              ;BRANCH IF NO
 2493
 2494                                  ;PART 2
 2495
 2496  015324  012700  000510                  MOV #510,R0             ;GET FIRST POSSIBLE VECTOR AREA
 2497  015330  012720  015504          T30L09: MOV #T30L08,(R0)+       ;SET UP VECTOR AREA TO HANDLE DOUBLE INTERRUPTS
 2498  015334  012720  000340                  MOV #340,(R0)+          ;SET PRIORITY = 7
 2499  015340  022700  001000                  CMP #1000,R0            ;AT END OF AREA?
 2500  015344  001371                          BNE T30L09              ;BRANCH IF NO
 2501  015346  016700  012436                  MOV BUFF1,R0            ;GET HIGHEST PRIORITY UBE ADDRESS POINTER
 2502  015352  016701  012434                  MOV BUFF1+2,R1          ;GET NEXT PRIORITY UBE ADDRESS POINTER
 2503  015356  012770  000002  000006  T30L14: MOV #2,@6(R0)           ;HAVE HIGHER PRIORITY UBE DO BR4
 2504  015364  012771  000004  000006          MOV #4,@6(R1)           ;HAVE NEXT LOWER ELEC. PRIORITY UBE DO BR5
 2505  015372  012770  015504  000014          MOV #T30L08,@14(R0)     ;SET UP HIGHER PRIORITY UBE VECTOR FOR DOUBLE INT.
 2506  015400  012771  015420  000014          MOV #T30L10,@14(R1)     ;SET UP FOR INTERRUPT FROM NEXT LOWER ELEC. PRIORITY UBE
 2507  015406  005277  165122          T30L11: INC @BEGO               ;START INTERRUPT
 2508  015412  005037  177776                  CLR @#PSW               ;ALLOW INTERRUPTS
 2509  015416  000001                          WAIT
 2510  015420  022626                  T30L10: CMP (SP)+,(SP)+         ;RESTORE STACK
 2511  015422  006371  000006                  ASL @6(R1)              ;HAVE NEXT PRIORITY UBE INT. ONE LEVEL HIGHER
 2512  015426  042771  000400  000006          BIC #400,@6(R1)         ;CLEAR SHIFTED RDY
 2513  015434  032771  000040  000006          BIT #40,@6(R1)          ;TESTED ALL BR LEVELS?
 2514  015442  001761                          BEQ T30L11              ;BRANCH IF NO
 2515  015444  020067  012350                  CMP R0,BUFF1+10         ;TESTED ALL UBE POSSIBLE?
 2516  015450  001511                          BEQ T30L12              ;BRANCH IF YES TO CLEAR BECR1 AND RESTORE TRAPS
 2517  015452  020067  012332                  CMP R0,BUFF1            ;JUST TESTED FIRST UBE?
 2518  015456  001005                          BNE T30L13              ;BRANCH IF NO
 2519  015460  016700  012326                  MOV BUFF1+2,R0          ;TEST SECOND HIGHEST PRIORITY UBE
 2520  015464  016701  012324                  MOV BUFF1+4,R1          ;GET THIRD HIGHEST PRIORITY UBE
 2521  015470  000732                          BR T30L14               ;GO TEST SECOND HIGHEST PRIORITY UBE
 2522  015472  016700  012316          T30L13: MOV BUFF1+4,R0          ;TEST THIRD HIGHEST PRIORITY UBE
 2523  015476  016701  012314                  MOV BUFF1+6,R1          ;GET FOURTH HIGHEST PRIORITY UBE
 2524  015502  000725                          BR T30L14               ;GO TEST THIRD HIGH PRIORITY UBE
 2525  015504  022626                  T30L08: CMP (SP)+,(SP)+         ;RESTORE STACK
 2526  015506  016067  000014  163500          MOV 14(R0),$REG0        ;SAVE INTERRUPT VECTOR OF BAD UBE
 2527  015514  012767  000004  163474          MOV #4,$REG1            ;SAVE BAD BR LEVEL
 2528  015522  016167  000014  163470          MOV 14(R1),$REG2        ;SAVE NEXT HIGHER PRIORITY UBE VECTOR
 2529  015530  032771  000004  000006          BIT #4,@6(R1)           ;WAS BR=5?
 2530  015536  001404                          BEQ T30L15              ;BRANCH IF NO
 2531  015540  012767  000005  163454          MOV #5,$REG3            ;BR=5
```

```
 2532  015546  000413                          BR T30L17               ;GO INDICATE ERROR
 2533  015550  032771  000010  000006  T30L15: BIT #10,@6(R1)          ;WAS BR=6?
 2534  015556  001404                          BEQ T30L16              ;BRANCH IF NO
 2535  015560  012767  000006  163434          MOV #6,$REG3            ;INDICATE BR=6
 2536  015566  000403                          BR T30L17               ;GO INDICATE ERROR
 2537  015570  012767  000007  163424  T30L16: MOV #7,$REG3            ;INDICATE BR=7
 2538  015576  104122                  T30L17: ERROR  D82              ;ERROR: TEST OF PASSING GRANTS FAILED
 2539  015600  032777  020000  163364          BIT #SW13,@SWR          ;INHIBIT ERROR TYPEOUTS?
 2540  015606  001032                          BNE T30L12              ;BRANCH IF YES
 2541  015610  104401  027156                  TYPE ,MSG7              ;TYPE FAILING UBE VECTOR
 2542  015614  016746  163374                  MOV     $REG0,-(SP)     ;;SAVE $REG0 FOR TYPEOUT
 2543  015620  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2544  015622  104401  027200                  TYPE ,MSG8              ;TYPE FAILING UBE BR LEVEL
 2545  015626  016746  163364                  MOV     $REG1,-(SP)     ;;SAVE $REG1 FOR TYPEOUT
 2546  015632  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2547  015634  104401  027215                  TYPE ,MSG9
 2548  015640  104401  027156                  TYPE ,MSG7              ;TYPE UBE USED TO TEST FAILING ONE
 2549  015644  016746  163350                  MOV     $REG2,-(SP)     ;;SAVE $REG2 FOR TYPEOUT
 2550  015650  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2551  015652  104401  027200                  TYPE ,MSG8              ;TYPE BR LEVEL TESTING
 2552  015656  016746  163340                  MOV     $REG3,-(SP)     ;;SAVE $REG3 FOR TYPEOUT
 2553  015662  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2554  015664  104401  027251                  TYPE ,MSG10
 2555  015670  004767  000534                  JSR PC,TERRPC           ;TYPE PC OF ERROR MSG
 2556  015674  012777  006003  164642  T30L12: MOV #6003,@BE1CR1       ;SETUP UBE TO DO A FUN3
 2557  015702  012777  006003  164652          MOV #6003,@BE2CR1       ;SETUP UBE TO DO A FUN3
 2558  015710  005767  164664                  TST BE3CR1              ;ARE THERE 3 UBE?
 2559  015714  001411                          BEQ 1$                  ;BRANCH IF NO
 2560  015716  012777  006003  164654          MOV #6003,@BE3CR1       ;SETUP UBE TO DO A FUN3
 2561  015724  005767  164666                  TST BE4CR1              ;ARE THERE 4 UBE?
 2562  015730  001403                          BEQ 1$                  ;BRANCH IF NO
 2563  015732  012777  006003  164656          MOV #6003,@BE4CR1       ;SETUP UBE TO DO A FUN3
 2564  015740  005037  177776          1$:     CLR @#PSW               ;ALLOW ALL UBE TO DO FUN3
 2565  015744  105777  164574          2$:     TSTB @BE1CR1            ;FIRST UBE DONE?
 2566  015750  100375                          BPL 2$                  ;BRANCH IF NO
 2567  015752  105777  164604          3$:     TSTB  @BE2CR1           ;SECOND UBE DONE?
 2568  015756  100375                          BPL 3$                  ;BRANCH IF NO
 2569  015760  005767  164614                  TST BE3CR1              ;ARE THERE THREE UBE?
 2570  015764  001411                          BEQ 6$                  ;BRANCH IF NO
 2571  015766  105777  164606          4$:     TSTB @BE3CR1            ;THIRD UBE DONE?
 2572  015772  100375                          BPL 4$                  ;BRANCH IF NO
 2573  015774  005767  164616                  TST BE4CR1              ;ARE THERE 4 UBE?
 2574  016000  001403                          BEQ 6$                  ;BRANCH IF NO
 2575  016002  105777  164610          5$:     TSTB @BE4CR1            ;FOURTH UBE DONE?
 2576  016006  100375                          BPL 5$                  ;BRANCH IF NO
 2577
 2578                                  ;RESTORE TRAP CATCHER
 2579
 2580  016010  012700  000510          6$:     MOV #510,R0             ;GET FIRST VECTOR ADDRESS
 2581  016014  012701  000512                  MOV #512,R1
 2582  016020  010120                  T30L20: MOV R1,(R0)+            ;PUT ADDRESS OF NEXT LOC IN THIS ONE
 2583  016022  005020                          CLR (R0)+               ;PUT HALT IN NEXT LOCATION
 2584  016024  022121                          CMP (R1)+,(R1)+         ;INC R1 BY 4
 2585  016026  020027  001000                  CMP R0,#1000            ;AT END OF VECTOR AREA?
 2586  016032  001372                          BNE T30L20              ;BRANCH IF NO
 2587  016034  005767  163072                  TST $PASS               ;FIRST PASS OF PROGRAM?
```

```
                        BNE $EOP                ;BRANCH IF NO
                        TYPE ,MSG2              ;ALL EXERCISORS TESTED
                                                ;NOTE:TO TEST PASSING OF GRANTS FOR
                                                ;THE LAST UBE,IT SHOULD BE
                                                ;SWAPPED WITH A UBE OF HIGHER
                                                ;ELECTRICAL PRIORITY


                .SBTTL  END OF PASS ROUTINE

                ;;*******************************************************************
                ;*INCREMENT THE PASS NUMBER ($PASS)
                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                ;*IF THERES A MONITOR GO TO IT
                ;*IF THERE ISN'T JUMP TO START1

016046          $EOP:
016046  000004          SCOPE
016050  005067  163060  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
016054  005067  163154  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
016060  005267  163046  INC     $PASS           ;;INCREMENT THE PASS NUMBER
016064  042767  100000  163040  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
016072  005327          DEC     (PC)+           ;;LOOP?
016074  000001  $EOPCT: .WORD   1
016076  003022          BGT     $DOAGN          ;;YES
016100  012737          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
016102  000001  $ENDCT: .WORD   1
016104  016074          $EOPCT
016106  104401  016153  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
016112  016746  163014  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
016116  104405          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
016120  104401  016150  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
016124  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
016130  001405          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
016132  000005          RESET                   ;;CLEAR THE WORLD
016134  004710  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
016136  000240          NOP                     ;;SAVE ROOM
016140  000240          NOP                     ;;FOR
016142  000240          NOP                     ;;ACT11
016144          $DOAGN:
016144  000137          JMP     @(PC)+          ;;RETURN
016146  003100  $RTNAD: .WORD   START1
016150     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
016153     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
016160  050040  051501  020123
016166  000043

                ;///////////////////////////////////////////////////////////////////
                ;SUBROUTINE TO CLEAR ALL UBE REG
                ;///////////////////////////////////////////////////////////////////
016170  005077  164334  CLRREG: CLR @BERE                 ;CLEAR ERROR CONDITIONS
016174  005077  164326          CLR @BECR2                ;CLEAR BECR2 REG
016200  005077  164320          CLR @BECR1                ;CLEAR BECR1 REG, EXCEPT RDY
016204  005077  164312          CLR @BEBA                 ;CLEAR BEBA REG
016210  005077  164304          CLR @BECC                 ;CLEAR BECC REG
016214  005077  164276          CLR @BEBD                 ;CLEAR BEBD REG
016220  000207                  RTS PC                    ;RETURN
```

```
                                ;////////////////////////////////////////////////////////////////////
                                ;SUBROUTINE TO RESTORE TRAP CATCHER TO UBE VECTOR AREA
                                ;////////////////////////////////////////////////////////////////////
016222  010546                RCATCH: MOV R5,-(SP)            ;SAVE R5 ON STACK
016224  016705  164302                MOV INTVEC,R5           ;GET INT. VECTOR
016230  005725                        TST(R5)+                ;CALC. INTVEC+2
016232  010577  164274                MOV R5,@INTVEC          ;PUT INTVEC+2 IN INTVEC
016236  005015                        CLR (R5)                ;PUT HALT IN INTVEC+2
016240  012605                        MOV (SP)+,R5            ;RESTORE R5
016242  000207                        RTS PC


                                ;////////////////////////////////////////////////////////////////////
                                ;SUBROUTINE TO CHECK IF RDY BIT SET
                                ;////////////////////////////////////////////////////////////////////
016244  005004                CRDY:   CLR R4
016246  005005                        CLR R5
016250  005205                2$:     INC R5                  ;UPDATE COUNT
016252  105777  164246                TSTB @BECR1             ;SEE IF RDY SET
016256  100405                        BMI 1$                  ;BRANCH IF SET
016260  032705  000200                BIT #200,R5             ;WAITED >100 MICROSECS?
016264  001771                        BEQ     2$              ;CONTINUE TO LOOK FOR RDY IF R5 NOT =128
016266  012704  000001                MOV     #1,R4           ;SET R4=1 TO INDICATE ERROR
016272  000207                1$:     RTS     PC              ;RETURN

                                ;////////////////////////////////////////////////////////////////////
                                ;SUBROUTINE TO DISREGARD UBE INTERRUPTS
                                ;////////////////////////////////////////////////////////////////////
016274  016705  164232          DINT:   MOV     INTVEC,R5       ;GET INTVEC AND
016300  005725                        TST     (R5)+           ;CALC. INTVEC+2
016302  010577  164224                MOV R5,@INTVEC          ;PUT ADDRESS OF NEXT LOC IN THIS ONE
016306  012715  000002                MOV     #2,(R5)         ;PUT AN RTI IN INTVEC+2
016312  000207                        RTS PC
                                ;////////////////////////////////////////////////////////////////////
                                ;SUBROUTINE TO RESTORE VECTOR AREA 0-56, 174, AND 176 FROM STACK AREA AND PUT TRAP CATCH
                                ;////////////////////////////////////////////////////////////////////
016314  016705  162704          RVEC:   MOV $TMP0,R5            ;GET AREA WHERE VECTOR STORED
016320  005004                        CLR R4                  ;SET R4 =TO FIRST LOC
016322  014524                1$:     MOV -(R5),(R4)+         ;RESTORE VECTORS
016324  022704  000060                CMP #60,R4              ;AT END OF AREA?
016330  001374                        BNE 1$                  ;BRANCH IF NO
016332  014537  000174                MOV     -(R5),  @#174   ;RESTORE SOFTWARE SWR
016336  014537  000176                MOV     -(R5),  @#176   ;
016342  012704  000060                MOV     #60,    R4      ;SET R4 FOR FIRST TRAP CATCHER
016346  012705  000062                MOV #62,R5              ;SET R5=TO FIRST TRAP CATCHER ADDRESS
016352  010524                2$:     MOV R5,(R4)+            ;PUT ADDRESS OF NEXT LOC IN THIS ONE
016354  005024                        CLR(R4)+                ;PUT HALT IN NEXT LOC
016356  022525                        CMP (R5)+,(R5)+         ;INC R5 BY 4
016360  022704  000174                CMP #174,R4             ;AT END OF  VECTOR AREA?
016364  001372                        BNE 2$                  ;BRANCH IF NO
016366  012704  000200                MOV     #200,   R4      ;AS ABOVE, PUT TRAP CATCHER IN AREA 200-776
016372  012705  000202                MOV     #202,   R5
016376  010524                3$:     MOV     R5,     (R4)+
016400  005024                        CLR     (R4)+
016402  022525                        CMP     (R5)+,  (R5)+
```

```
                                        CMP     #1000,  R4
                                        BNE     3$
                                        MOV #137,@#200          ;RESTORE JMP @#START TO LOC 200
                                        MOV #START,@#202        ;
                                        RTS PC                  ;RETURN
                                ;///////////////////////////////////////////////////////////////////////
                                ;SUBROUTINE TO TYPE PC OF ERROR MESSAGE
                                ;///////////////////////////////////////////////////////////////////////
                                TERRPC: BIT #SW13,@SWR          ;INHIBITS ERROR TYPOUTS?
                                        BNE 1$                  ;BRANCH IF YES
                                        TYPE ,MSG15             ;PC OF ERROR MSG WAS:
                                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                        TYPE    ,MSG17          ;TEST NUMBER WAS:
                                        MOV     $TSTNM, -(SP)   ;SAVE $TSTNM FOR TYPEOUT
                                        TYPOS                   ;GO TYPE -OCTAL ASCII
                                        .BYTE   2               ;TYPE 2 DIGITS
                                        .BYTE   0               ;SUPPRESS LEADING ZEROS
                                1$:     RTS PC

                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;*********************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

                                $SCOPE:
                                1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
                                        BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
                                $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
                                        MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
                                        MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
                                        TST     @#177060        ;;TIME OUT ON XOR?
                                        MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
                                        BR      $SVLAD          ;;GO TO THE NEXT TEST
                                5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
                                        MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
                                        BR      7$              ;;LOOP ON THE PRESENT TEST
                                6$:;#####END OF CODE FOR THE XOR TESTER#####
                                2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
                                        BEQ     3$              ;;BR IF NO
                                        CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
                                        BHI     3$              ;;BR IF NO
                                        BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
                                        BEQ     4$              ;;BR IF NO
                                7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
                                        BR      $OVER
```

```
 2756  016574  105067  162335          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 2757  016600  005067  162430                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 2758  016604  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
 2759  016606  032777  004000  162356  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 2760  016614  001011                          BNE     1$              ;;BR IF YES
 2761  016616  005767  162310                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 2762  016622  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
 2763  016624  005267  162306                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
 2764  016630  026767  162400  162300          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 2765  016636  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 2766  016640  012767  000001  162270  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 2767  016646  016767  000044  162360          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 2768  016654  105267  162254          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 2769  016660  011667  162254                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 2770  016664  011667  162252                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 2771  016670  005067  162342                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 2772  016674  112767  000001  162245          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 2773  016702  016777  162226  162264  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 2774  016710  016716  162224                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 2775  016714  000002                          RTI                     ;;FIXES PS
 2776  016716  000012          $MXCNT: 10.                             ;;MAX. NUMBER OF ITERATIONS
 2777                          .SBTTL  ERROR HANDLER ROUTINE
 2778
 2779                          ;;**************************************************************
 2780                          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 2781                          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 2782                          ;*AND GO TO $ERRTYP ON ERROR
 2783                          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 2784                          ;*SW15=1        HALT ON ERROR
 2785                          ;*SW13=1        INHIBIT ERROR TYPEOUTS
 2786                          ;*SW10=1        BELL ON ERROR
 2787                          ;*SW09=1        LOOP ON ERROR
 2788                          ;*CALL
 2789                          ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 2790
 2791  016720          $ERROR:
 2792  016720  105267  162211          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 2793  016724  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 2794  016726  016777  162202  162240          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 2795  016734  032777  002000  162230          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 2796  016742  001402                          BEQ     1$              ;;NO - SKIP
 2797  016744  104401  001240                  TYPE    ,$BELL          ;;RING BELL
 2798  016750  005267  162170          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 2799  016754  011667  162170                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 2800  016760  162767  000002  162162          SUB     #2,$ERRPC
 2801  016766  117767  162156  162152          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 2802  016774  032777  020000  162170          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 2803  017002  001004                          BNE     20$             ;;SKIP TYPEOUTS
 2804  017004  004767  000056                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
 2805  017010  104401  001245                  TYPE    ,$CRLF
 2806  017014                          20$:
 2807  017014  005777  162152          2$:     TST     @SWR            ;;HALT ON ERROR
 2808  017020  100001                          BPL     3$              ;;SKIP IF CONTINUE
 2809  017022  000000                          HALT                    ;;HALT ON ERROR!
 2810  017024  032777  001000  162140  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 2811  017032  001402                          BEQ     4$              ;;BR IF NO
```

```
                        MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
4$:                     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
                        BEQ     5$              ;;BR IF NONE
                        MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
5$:
                        CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
                        BNE     6$              ;;BRANCH IF NO
                        HALT                    ;;YES
6$:
                        RTI                     ;;RETURN
                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

$ERRTYP:
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
                        MOV     R0,-(SP)        ;;SAVE R0
                        CLR     R0              ;;PICKUP THE ITEM INDEX
                        BISB    @#$ITEMB,R0
                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                ;;TYPE THE PC OF THE ERROR
                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                ;;ERROR ADDRESS
                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                        BR      6$              ;;GET OUT
1$:                     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
                        ASL     R0
                        ASL     R0
                        ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
                        MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
2$:                     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3$:                     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
                        BEQ     5$              ;;SKIP TYPEOUT IF 0
                        TYPE                    ;;TYPE THE "DATA HEADER"
4$:                     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
5$:                     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
                        BNE     7$              ;;GO TYPE THE DATA
6$:                     MOV     (SP)+,R0        ;;RESTORE R0
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
                        RTS     PC              ;;RETURN
7$:
                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
                        BEQ     6$              ;;BR IF NO
                        TYPE    ,8$             ;;TYPE TWO(2) SPACES
                        BR      7$              ;;LOOP
8$:                     .ASCIZ  /  /            ;;TWO(2) SPACES
```

```
                        017222                          .EVEN
                                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                                ;;**********************************************************************
                                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                                ;*REPLACED WITH SPACES.
                                                ;*CALL:
                                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                                ;*      TYPDS                   ;;GO TO THE ROUTINE

017222                                          $TYPDS:
017222  010046                                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
017224  010146                                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
017226  010246                                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
017230  010346                                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
017232  010546                                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
017234  012746  020200                               MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
017240  016605  000020                               MOV     20(SP),R5       ;;GET THE INPUT NUMBER
017244  100004                                       BPL     1$              ;;BR IF INPUT IS POS.
017246  005405                                       NEG     R5              ;;MAKE THE BINARY NUMBER POS.
017250  112766  000055  000001                       MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
017256  005000                                1$:    CLR     R0              ;;ZERO THE CONSTANTS INDEX
017260  012703  017436                               MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
017264  112723  000040                               MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
017270  005002                                2$:    CLR     R2              ;;CLEAR THE BCD NUMBER
017272  016001  017426                               MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
017276  160105                                3$:    SUB     R1,R5           ;;FORM THIS BCD DIGIT
017300  002402                                       BLT     4$              ;;BR IF DONE
017302  005202                                       INC     R2              ;;INCREASE THE BCD DIGIT BY 1
017304  000774                                       BR      3$
017306  060105                                4$:    ADD     R1,R5           ;;ADD BACK THE CONSTANT
017310  005702                                       TST     R2              ;;CHECK IF BCD DIGIT=0
017312  001002                                       BNE     5$              ;;FALL THROUGH IF 0
017314  105716                                       TSTB    (SP)            ;;STILL DOING LEADING 0'S?
017316  100407                                       BMI     7$              ;;BR IF YES
017320  106316                                5$:    ASLB    (SP)            ;;MSD?
017322  103003                                       BCC     6$              ;;BR IF NO
017324  116663  000001  177777                       MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
017332  052702  000060                        6$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
017336  052702  000040                        7$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
017342  110223                                       MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
017344  005720                                       TST     (R0)+           ;;JUST INCREMENTING
017346  020027  000010                               CMP     R0,#10          ;;CHECK THE TABLE INDEX
017352  002746                                       BLT     2$              ;;GO DO THE NEXT DIGIT
017354  003002                                       BGT     8$              ;;GO TO EXIT
017356  010502                                       MOV     R5,R2           ;;GET THE LSD
017360  000764                                       BR      6$              ;;GO CHANGE TO ASCII
017362  105726                                8$:    TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
017364  100003                                       BPL     9$              ;;BR IF NO
017366  116663  177777  177776                       MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
017374  105013                                9$:    CLRB    (R3)            ;;SET THE TERMINATOR
017376  012605                                       MOV     (SP)+,R5        ;;POP STACK INTO R5
017400  012603                                       MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
 2924  017402  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 2925  017404  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 2926  017406  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 2927  017410  104401  017436                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 2928  017414  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 2929  017422  012616                          MOV     (SP)+,(SP)
 2930  017424  000002                          RTI                     ;;RETURN TO USER
 2931  017426  023420                $DTBL:    10000.
 2932  017430  001750                          1000.
 2933  017432  000144                          100.
 2934  017434  000012                          10.
 2935  017436  000004                $DBLK:    .BLKW   4
 2936                                .SBTTL  TYPE ROUTINE
 2937
 2938                                ;;******************************************************************
 2939                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 2940                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 2941                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 2942                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 2943                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 2944                                ;*
 2945                                ;*CALL:
 2946                                ;*1) USING A TRAP INSTRUCTION
 2947                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 2948                                ;*OR
 2949                                ;*      TYPE
 2950                                ;*      MESADR
 2951                                ;*
 2952
 2953  017446  105767  161537        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 2954  017452  100002                        BPL     1$              ;;BR IF YES
 2955  017454  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
 2956  017456  000407                        BR      3$              ;;LEAVE
 2957  017460  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
 2958  017462  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 2959  017466  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 2960  017470  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 2961  017472  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 2962  017474  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
 2963  017476  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 2964  017502  000002                        RTI                     ;;RETURN
 2965  017504  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF .<HT>
 2966  017510  001430                        BEQ     8$
 2967  017512  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 2968  017516  001006                        BNE     5$
 2969  017520  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
 2970  017522  104401                        TYPE                    ;;TYPE A CR AND LF
 2971  017524  001245                        $CRLF
 2972  017526  105067  000130                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 2973  017532  000755                        BR      2$              ;;GET NEXT CHARACTER
 2974  017534  004767  000056        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 2975  017540  126726  161444        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 2976  017544  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
 2977  017546  016746  161434                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 2978                                                                ;;AND THE NULL CHAR.
 2979  017552  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
```

```
 2980  017556  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 2981  017560  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
 2982  017564  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 2983  017570  000770                          BR      7$              ;;LOOP
 2984
 2985                                  ;HORIZONTAL TAB PROCESSOR
 2986
 2987  017572  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 2988  017576  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 2989  017602  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 2990  017610  001372                          BNE     9$              ;;TAB STOP
 2991  017612  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
 2992  017614  000724                          BR      2$              ;;GET NEXT CHARACTER
 2993  017616  105777  161360          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 2994  017622  100375                          BPL     $TYPEC
 2995  017624  116677  000002  161352          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 2996  017632  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 2997  017640  001003                          BNE     1$              ;;BRANCH IF NO
 2998  017642  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 2999  017646  000406                          BR      $TYPEX          ;;EXIT
 3000  017650  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 3001  017656  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
 3002  017660  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
 3003  017662  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 3004  017664  000207                  $TYPEX: RTS     PC
 3005
 3006                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 3007
 3008                                  ;;**********************************************************************
 3009                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 3010                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 3011                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 3012                                  ;*CALL:
 3013                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 3014                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
 3015                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 3016                                  ;*      .BYTE   M               ;;M=1 OR 0
 3017                                  ;*                                      ;;1=TYPE LEADING ZEROS
 3018                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
 3019                                  ;*
 3020                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 3021                                  ;*$TYPOS OR $TYPOC
 3022                                  ;*CALL:
 3023                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 3024                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
 3025                                  ;*
 3026                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 3027                                  ;*CALL:
 3028                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 3029                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
 3030
 3031  017666  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 3032  017672  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 3033  017700  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 3034  017704  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 3035  017710  000406                          BR      $TYPON
```

```
3036  017712  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3037  017720  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3038  017726  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3039  017734  010346                          MOV     R3,-(SP)        ;;SAVE R3
3040  017736  010446                          MOV     R4,-(SP)        ;;SAVE R4
3041  017740  010546                          MOV     R5,-(SP)        ;;SAVE R5
3042  017742  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3043  017746  005404                          NEG     R4
3044  017750  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3045  017754  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3046  017760  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3047  017764  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3048  017770  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
3049  017772  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3050  017774  000404                          BR      3$              ;;GO DO MSB
3051  017776  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
3052  020000  006105                          ROL     R5
3053  020002  006105                          ROL     R5
3054  020004  010503                          MOV     R5,R3
3055  020006  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3056  020010  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
3057  020014  100016                          BPL     7$              ;;BR IF NO
3058  020016  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
3059  020022  001002                          BNE     4$              ;;TEST FOR 0
3060  020024  005704                          TST     R4              ;;SUPPRESS THIS 0?
3061  020026  001403                          BEQ     5$              ;;BR IF YES
3062  020030  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3063  020032  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3064  020036  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3065  020042  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
3066  020046  104401  020106                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
3067  020052  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
3068  020056  003347                          BGT     2$              ;;BR IF MORE TO DO
3069  020060  002402                          BLT     6$              ;;BR IF DONE
3070  020062  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3071  020064  000744                          BR      2$              ;;GO DO THE LAST DIGIT
3072  020066  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
3073  020070  012604                          MOV     (SP)+,R4        ;;RESTORE R4
3074  020072  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3075  020074  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3076  020102  012616                          MOV     (SP)+,(SP)
3077  020104  000002                          RTI                     ;;RETURN
3078  020106     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3079  020107     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3080  020110     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3081  020111     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3082  020112  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3083                                  .SBTTL  TRAP DECODER
3084
3085                                  ;;**************************************************************************
3086                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3087                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3088                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3089                                  ;*GO TO THAT ROUTINE.
3090
3091  020114  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
```

```
3092  020116  016600  000002              MOV     2(SP),R0        ;;GET TRAP ADDRESS
3093  020122  005740                      TST     -(R0)           ;;BACKUP BY 2
3094  020124  111000                      MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
3095  020126  006300                      ASL     R0              ;;POSITION FOR INDEXING
3096  020130  016000  020150              MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
3097  020134  000200                      RTS     R0              ;;GO TO ROUTINE
3098
3099
3100                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3101
3102  020136  011646                $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
3103  020140  016666  000004  000002        MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
3104  020146  000002                      RTI                     ;;RESTORE THE PSW
3105
3106                                .SBTTL  TRAP TABLE
3107
3108                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3109                                ;*BY THE "TRAP" INSTRUCTION.
3110
3111                                ;       ROUTINE
3112                                ;       -------
3113  020150  020136                $TRPAD: .WORD   $TRAP2
3114  020152  017446                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
3115  020154  017712                        $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3116  020156  017666                        $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3117  020160  017726                        $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3118  020162  017222                        $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
3119
3120
3121                                .SBTTL  POWER DOWN AND UP ROUTINES
3122
3123                                ;;**************************************************************
3124                                ;POWER DOWN ROUTINE
3125  020164  012737  020324  000024 $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
3126  020172  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
3127  020200  010046                       MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3128  020202  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3129  020204  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3130  020206  010346                       MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3131  020210  010446                       MOV     R4,-(SP)        ;;PUSH R4 ON STACK
3132  020212  010546                       MOV     R5,-(SP)        ;;PUSH R5 ON STACK
3133  020214  017746  160752               MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
3134  020220  010667  000104               MOV     SP,$SAVR6       ;;SAVE SP
3135  020224  012737  020236  000024       MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
3136  020232  000000                       HALT
3137  020234  000776                       BR      .-2             ;;HANG UP
3138
3139                                ;;**************************************************************
3140                                ;POWER UP ROUTINE
3141  020236  012737  020324  000024 $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
3142  020244  016706  000060               MOV     $SAVR6,SP       ;;GET SP
3143  020250  005067  000054               CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
3144  020254  005267  000050        1$:    INC     $SAVR6          ;;WAIT FOR THE INC
3145  020260  001375                       BNE     1$              ;;OF  WORD
3146  020262  012677  160704               MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
3147  020266  012605                       MOV     (SP)+,R5        ;;POP STACK INTO R5
```

```
  3148  020270  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
  3149  020272  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
  3150  020274  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
  3151  020276  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
  3152  020300  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
  3153  020302  012737  020164  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
  3154  020310  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
  3155  020316  104401                          TYPE                    ;REPORT THE POWER FAILURE
  3156  020320  020332                $PWRMG:   .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
  3157  020322  000002                          RTI
  3158  020324  000000                $ILLUP:   HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
  3159  020326  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
  3160  020330  000000                $SAVR6:   0                       ;;PUT THE SP HERE
  3161  020332  005015  047520  042527  $POWER:  .ASCIZ  <15><12>"POWER"
  3162  020340  000122
  3163                                          .EVEN
  3164                                ;;*****************************************************************
  3165                                ;;*****************************************************************
  3166  020342  005015  046505  050101  MSG1:    .ASCIZ<15><12>/EMAP: /
  3167  020350  020072     000
  3168  020353     015  040412  046114  MSG2:    .ASCII<15><12>/ALL EXERCISORS TESTED/<15><12>
  3169  020360  042440  042530  041522
  3170  020366  051511  051117  020123
  3171  020374  042524  052123  042105
  3172  020402  005015
  3173  020404  020040  047040  052117          .ASCII/   NOTE:TO TEST PASSING OF GRANTS FOR THE LAST UBE/<15><12>
  3174  020412  035105  047524  052040
  3175  020420  051505  020124  040520
  3176  020426  051523  047111  020107
  3177  020434  043117  043440  040522
  3178  020442  052116  020123  047506
  3179  020450  020122  044124  020105
  3180  020456  040514  052123  052440
  3181  020464  042502  005015
  3182  020470  020040  020040  020040          .ASCII/         IT SHOULD BE SWAPPED WITH A UBE/<15><12>
  3183  020476  020040  052111  051440
  3184  020504  047510  046125  020104
  3185  020512  042502  051440  040527
  3186  020520  050120  042105  053440
  3187  020526  052111  020110  020101
  3188  020534  041125  006505     012
  3189  020541     040  020040  020040          .ASCIZ/         OF HIGHER ELECTRICAL PRIORITY/<15><12>
  3190  020546  020040  047440  020106
  3191  020554  044510  044107  051105
  3192  020562  042440  042514  052103
  3193  020570  044522  040503  020114
  3194  020576  051120  047511  044522
  3195  020604  054524  005015     000
  3196  020611     015  041012  051525  MSG5:    .ASCIZ<15><12>*BUS PARITY NOT TESTED ON 11/05 OR 11/20 MACHINES*<15><12>
  3197  020616  050040  051101  052111
  3198  020624  020131  047516  020124
  3199  020632  042524  052123  042105
  3200  020640  047440  020116  030461
  3201  020646  030057  020065  051117
  3202  020654  030440  027461  030062
  3203  020662  046440  041501  044510
```

```
3204  020670  042516  006523  000012
3205  020676  047516  051040  051505  EM1:    .ASCIZ/NO RESPONSE TO REG ADDRESSES OR NO DEVICE PRESENT/
3206  020704  047520  051516  020105
3207  020712  047524  051040  043505
3208  020720  040440  042104  042522
3209  020726  051523  051505  047440
3210  020734  020122  047516  042040
3211  020742  053105  041511  020105
3212  020750  051120  051505  047105
3213  020756  000124
3214  020760  040506  040524  020114  EM2.    .ASCIZ/FATAL ERROR:REG FAILED TO CLEAR/
3215  020766  051105  047522  035122
3216  020774  042522  020107  040506
3217  021002  046111  042105  052040
3218  021010  020117  046103  040505
3219  021016  000122
3220  021020  042522  020107  042101  DH2.    .ASCIZ*REG ADD/REG CONTENTS *
3221  021026  027504  042522  020107
3222  021034  047503  052116  047105
3223  021042  051524  000040
3224                                  .EVEN
3225  021046  001214  001216  000000  DT2:    .WORD $REG0,$REG1,0
3226  021054  040506  040524  020114  EM3:    .ASCIZ/FATAL ERROR:CPU DID NOT RECEIVE SSYN/
3227  021062  051105  047522  035122
3228  021070  050103  020125  044504
3229  021076  0<0104  047516  020124
3230  021104  042522  042503  053111
3231  021112  020105  051523  047131
3232  021120     000
3233  021121     120  020103  040527  DH3:    .ASCIZ/PC WAS/
3234  021126  000123
3235                                  .EVEN
3236  021130  001214  000000          DT3:    .WORD $REG0,0
3237  021134  040506  040524  020114  EM4:    .ASCIZ/FATAL ERROR:REG FAILED TO FLOAT A '1'/
3238  021142  051105  047522  035122
3239  021150  042522  020107  040506
3240  021156  046111  042105  052040
3241  021164  020117  046106  040517
3242  021172  020124  020101  030447
3243  021200  000047
3244  021202  042522  020107  042101  DH4:    .ASCIZ*REG ADD/DATA IS/DATA SHOULD BE*
3245  021210  027504  040504  040524
3246  021216  044440  027523  040504
3247  021224  040524  051440  047510
3248  021232  046125  020104  042502
3249  021240     000
3250          021242                  .EVEN
3251  021242  001214  001216  001220  DT4:    .WORD $REG0,$REG1,$REG2,0
3252  021250  000000
3253  021252  040506  040524  020114  EM5:    .ASCIZ/FATAL ERROR:REG FAILED TO FLOAT A '0'/
3254  021260  051105  047522  035122
3255  021266  042522  020107  040506
3256  021274  046111  042105  052040
3257  021302  020117  046106  040517
3258  021310  020124  020101  030047
3259  021316  000047
```

```
 3260  021320  040506  040524  020114  EM6:    .ASCIZ/FATAL ERROR:CONTROL REG HELD WRONG DATA/
 3261  021326  051105  047522  035122
 3262  021334  047503  052116  047522
 3263  021342  020114  042522  020107
 3264  021350  042510  042114  053440
 3265  021356  047522  043516  042040
 3266  021364  052101  000101
 3267  021370  040506  040524  020114  EM7:    .ASCIZ/FATAL ERROR:DUAL ADDRESSING ERROR/
 3268  021376  051105  047522  035122
 3269  021404  052504  046101  040440
 3270  021412  042104  042522  051523
 3271  021420  047111  020107  051105
 3272  021426  047522  000122
 3273  021432  042522  020107  042101  DH7:    .ASCIZ*REG ADD/REG ADD WERE SIMULATANEOUSLY WRITTEN*
 3274  021440  027504  042522  020107
 3275  021446  042101  020104  042527
 3276  021454  042522  051440  046511
 3277  021462  046125  052101  047101
 3278  021470  047505  051525  054514
 3279  021476  053440  044522  052124
 3280  021504  047105     000
 3281          021510                  .EVEN
 3282  021510  001214  001216  000000  DT7:    .WORD $REG0,$REG1,0
 3283  021516  051105  047522  035122  EM8:    .ASCIZ/ERROR: SETTING PB PARITY FAILED TO CAUSE CPU TO TRAP/
 3284  021524  051440  052105  044524
 3285  021532  043516  050040  020102
 3286  021540  040520  044522  054524
 3287  021546  043040  044501  042514
 3288  021554  020104  047524  041440
 3289  021562  052501  042523  041440
 3290  021570  052520  052040  020117
 3291  021576  051124  050101     000
 3292  021603     105  051122  051117  EM9:    .ASCIZ/ERROR: GO BIT FAILED TO LOAD '1'/
 3293  021610  020072  047507  041040
 3294  021616  052111  043040  044501
 3295  021624  042514  020104  047524
 3296  021632  046040  040517  020104
 3297  021640  030447  000047
 3298  021644  051105  047522  035122  EM10:   .ASCIZ/ERROR: GO BIT FAILED TO LOAD '0'/
 3299  021652  043440  020117  044502
 3300  021660  020124  040506  046111
 3301  021666  042105  052040  020117
 3302  021674  047514  042101  023440
 3303  021702  023460     000
 3304  021705     106  052101  046101  EM11:   .ASCIZ/FATAL ERROR: GO BIT FAILED TO CLEAR/
 3305  021712  042440  051122  051117
 3306  021720  020072  047507  041040
 3307  021726  052111  043040  044501
 3308  021734  042514  020104  047524
 3309  021742  041440  042514  051101
 3310  021750     000
 3311  021751     106  052101  046101  EM12:   .ASCIZ/FATAL ERROR: READY BIT FAILED TO SET/
 3312  021756  042440  051122  051117
 3313  021764  020072  042522  042101
 3314  021772  020131  044502  020124
 3315  022000  040506  046111  042105
```

```
3316  022006  052040  020117  042523
3317  022014  000124
3318  022016  047524  041440  042514  EM14:   .ASCIZ/TO CLEAR BIT 10 OF BECR1/
3319  022024  051101  041040  052111
3320  022032  030440  020060  043117
3321  022040  041040  041505  030522
3322  022046     000
3323  022047     105  051122  051117  EM15:   .ASCIZ/ERROR: ERROR BITS IN BECR2 SET WHEN SHOULD BE CLEAR/
3324  022054  020072  051105  047522
3325  022062  020122  044502  051524
3326  022070  044440  020116  042502
3327  022076  051103  020062  042523
3328  022104  020124  044127  047105
3329  022112  051440  047510  046125
3330  022120  020104  042502  041440
3331  022126  042514  051101     000
3332  022133     103  047117  042524  DH15:   .ASCIZ/CONTENTS OF BECR2/
3333  022140  052116  020123  043117
3334  022146  041040  041505  031122
3335  022154     000
3336  022155     106  052101  046101  EM16:   .ASCIZ/FATAL ERROR: READY BIT FAILED TO CLEAR OR GO FAILED TO SET/
3337  022162  042440  051122  051117
3338  022170  020072  042522  042101
3339  022176  020131  044502  020124
3340  022204  040506  046111  042105
3341  022212  052040  020117  046103
3342  022220  040505  020122  051117
3343  022226  043440  020117  040506
3344  022234  046111  042105  052040
3345  022242  020117  042523  000124
3346  022250  051105  047522  035122  EM17:   .ASCIZ/ERROR: UBE FAILED TO INTERRUPT/
3347  022256  052440  042502  043040
3348  022264  044501  042514  020104
3349  022272  047524  044440  052116
3350  022300  051105  052522  052120
3351  022306     000
3352  022307     102  020122  051511  DH17:   .ASCIZ*BR IS  / PRIORITY IS*
3353  022314  020040  020057  051120
3354  022322  047511  044522  054524
3355  022330  044440  000123
3356  022334  051105  047522  035122  EM18:   .ASCIZ/ERROR: UBE INTERRUPTED WHEN PSW AT SAME PRIORITY LEVEL/
3357  022342  052440  042502  044440
3358  022350  052116  051105  052522
3359  022356  052120  042105  053440
3360  022364  042510  020116  051520
3361  022372  020127  052101  051440
3362  022400  046501  020105  051120
3363  022406  047511  044522  054524
3364  022414  046040  053105  046105
3365  022422     000
3366  022423     125  042502  041040  DH18:   .ASCIZ/UBE BR WAS/
3367  022430  020122  040527  000123
3368  022436  051105  047522  035122  EM19:   .ASCIZ/ERROR: UBE FALSELY INTERRUPTED AT A HIGHER LEVEL/
3369  022444  052440  042502  043040
3370  022452  046101  042523  054514
3371  022460  044440  052116  051105
```

```
 3372  022466  052522  052120  042105
 3373  022474  040440  020124  020101
 3374  022502  044510  044107  051105
 3375  022510  046040  053105  046105
 3376  022516     000
 3377  022517     110  043511  042510  DH19:   .ASCIZ/HIGHER LEVEL WAS/
 3378  022524  020122  042514  042526
 3379  022532  020114  040527  000123
 3380  022540  051105  047522  035122  EM20:   .ASCIZ/ERROR: UBE INTERRUPTED TO WRONG VECTOR/
 3381  022546  052440  042502  044440
 3382  022554  052116  051105  052522
 3383  022562  052120  042105  052040
 3384  022570  020117  051127  047117
 3385  022576  020107  042526  052103
 3386  022604  051117     000
 3387
 3388  022607     105  051122  051117  EM21:   .ASCIZ/ERROR: SIMULTANEOUS GO FAILED/
 3389  022614  020072  044523  052515
 3390  022622  052114  047101  047505
 3391  022630  051525  043440  020117
 3392  022636  040506  046111  042105
 3393  022644     000
 3394  022645     105  051122  051117  EM22:   .ASCIZ/ERROR: NO, NO SACK BIT FALSELY SET/
 3395  022652  020072  047516  020054
 3396  022660  047516  051440  041501
 3397  022666  020113  044502  020124
 3398  022674  040506  051514  046105
 3399  022702  020131  042523  000124
 3400  022710  051105  047522  035122  EM23:   .ASCIZ/ERROR: NO INT. SSYN BIT FALSELY SET/
 3401  022716  047040  020117  047111
 3402  022724  027124  051440  054523
 3403  022732  020116  044502  020124
 3404  022740  040506  051514  046105
 3405  022746  020131  042523  000124
 3406  022754  051105  047522  035122  EM24:   .ASCIZ/ERROR: DATI FAILED TO LOAD PROPER DATA/
 3407  022762  042040  052101  020111
 3408  022770  040506  046111  042105
 3409  022776  052040  020117  047514
 3410  023004  042101  050040  047522
 3411  023012  042520  020122  040504
 3412  023020  040524     000
 3413  023023     102  041105  020104  DH24:   .ASCIZ*BEBD /MEM DATA/MEM ADD/DATA SHOULD BE IN MEM*
 3414  023030  046457  046505  042040
 3415  023036  052101  027501  042515
 3416  023044  020115  042101  027504
 3417  023052  040504  040524  051440
 3418  023060  047510  046125  020104
 3419  023066  042502  044440  020116
 3420  023074  042515  000115
 3421                                  .EVEN
 3422  023100  001214  001216  001220  DT24:   .WORD   $REG0,$REG1,$REG2,$REG3,0
 3423  023106  001222  000000
 3424  023112  040504  047524  043040  EM25:   .ASCIZ/DATO FAILED TO LOAD PROPER DATA/
 3425  023120  044501  042514  020104
 3426  023126  047524  046040  040517
 3427  023134  020104  051120  050117
```

```
 3428  023142  051105  042040  052101
 3429  023150  000101
 3430  023152  040504  044524  020120  EM26:   .ASCIZ/DATIP FAILED TO LOAD PROPER DATA/
 3431  023160  040506  046111  042105
 3432  023166  052040  020117  047514
 3433  023174  042101  050040  047522
 3434  023202  042520  020122  040504
 3435  023210  040524     000
 3436  023213     104  052101  041117  EM27:   .ASCIZ/DATOB FILED TO LOAD PROPER DATA/
 3437  023220  043040  046111  042105
 3438  023226  052040  020117  047514
 3439  023234  042101  050040  047522
 3440  023242  042520  020122  040504
 3441  023250  040524     000
 3442  023253     104  052101  020111  EM28:   .ASCIZ/DATI FAILED TO SET RDY/
 3443  023260  040506  046111  042105
 3444  023266  052040  020117  042523
 3445  023274  020124  042122  000131
 3446  023302  040504  047524  043040  EM29:   .ASCIZ/DATO FAILED TO SET RDY/
 3447  023310  044501  042514  020104
 3448  023316  047524  051440  052105
 3449  023324  051040  054504     000
 3450  023331     104  052101  050111  EM30:   .ASCIZ/DATIP FAILED TO SET RDY/
 3451  023336  043040  044501  042514
 3452  023344  020104  047524  051440
 3453  023352  052105  051040  054504
 3454  023360     000
 3455  023361     104  052101  041117  EM31:   .ASCIZ/DATOB FAILED TO SET RDY/
 3456  023366  043040  044501  042514
 3457  023374  020104  047524  051440
 3458  023402  052105  051040  054504
 3459  023410     000
 3460  023411     105  051122  051117  EM32:   .ASCIZ/ERROR: INH. DATA SHIFT ON DATIP FAILED/
 3461  023416  020072  047111  027110
 3462  023424  042040  052101  020101
 3463  023432  044123  043111  020124
 3464  023440  047117  042040  052101
 3465  023446  050111  043040  044501
 3466  023454  042514  000104
 3467  023460  051105  047522  035122  EM33:   .ASCIZ/ERROR: DATOB ON DATIP FAILED/
 3468  023466  042040  052101  041117
 3469  023474  047440  020116  040504
 3470  023502  044524  020120  040506
 3471  023510  046111  042105     000
 3472  023515     105  051122  051117  EM34:   .ASCIZ/ERROR: UBE DID DATI FROM WRONG LOCATION/
 3473  023522  020072  041125  020105
 3474  023530  044504  020104  040504
 3475  023536  044524  043040  047522
 3476  023544  020115  051127  047117
 3477  023552  020107  047514  040503
 3478  023560  044524  047117     000
 3479  023565     115  046505  046040  DH34:   .ASCIZ/MEM LOC WANTED/
 3480  023572  041517  053440  047101
 3481  023600  042524  000104
 3482  023604  051105  047522  035122  EM35:   .ASCIZ/ERROR: BEBA LOWER 16 BITS DID NOT COUNT BY 2/
 3483  023612  041040  041105  020101
```

```
 3484  023620  047514  042527  020122
 3485  023626  033061  041040  052111
 3486  023634  020123  044504  020104
 3487  023642  047516  020124  047503
 3488  023650  047125  020124  054502
 3489  023656  031040     000
 3490  023661     050  042522  024507  DH35:   .ASCIZ*(REG) /DATA SHOULD BE*
 3491  023666  027440  040504  040524
 3492  023674  051440  047510  046125
 3493  023702  020104  042502     000
 3494  023707     105  051122  051117  EM36:   .ASCIZ/ERROR: BEBA BIT A16,17 DID NOT COUNT=0/
 3495  023714  020072  042502  040502
 3496  023722  041040  052111  040440
 3497  023730  033061  030454  020067
 3498  023736  044504  020104  047516
 3499  023744  020124  047503  047125
 3500  023752  036524  000060
 3501  023756  030501  026066  030501  DH36:   .ASCIZ/A16,A17/
 3502  023764  000067
 3503  023766  051105  047522  035122  EM37:   .ASCIZ/ERROR: BEBA DID NOT COUNT BY 1/
 3504  023774  041040  041105  020101
 3505  024002  044504  020104  047516
 3506  024010  020124  047503  047125
 3507  024016  020124  054502  030440
 3508  024024     000
 3509  024025     105  051122  051117  EM38:   .ASCIZ/ERROR: INTERRUPT FAILED TO UPDATE BECC TO CORRECT VALUE/
 3510  024032  020072  047111  042524
 3511  024040  051122  050125  020124
 3512  024046  040506  046111  042105
 3513  024054  052040  020117  050125
 3514  024062  040504  042524  041040
 3515  024070  041505  020103  047524
 3516  024076  041440  051117  042522
 3517  024104  052103  053040  046101
 3518  024112  042525     000
 3519  024115     105  051122  051117  EM39:   .ASCIZ/ERROR: BEBA INCREMENTED WHEN IT WAS INHIBITED/
 3520  024122  020072  042502  040502
 3521  024130  044440  041516  042522
 3522  024136  042515  052116  042105
 3523  024144  053440  042510  020116
 3524  024152  052111  053440  051501
 3525  024160  044440  044116  041111
 3526  024166  052111  042105     000
 3527  024173     105  051122  051117  EM40:   .ASCIZ/ERROR: BECC INCREMENTED WHEN IT WAS INHIBITED/
 3528  024200  020072  042502  041503
 3529  024206  044440  041516  042522
 3530  024214  042515  052116  042105
 3531  024222  053440  042510  020116
 3532  024230  052111  053440  051501
 3533  024236  044440  044116  041111
 3534  024244  052111  042105     000
 3535  024251     105  051122  051117  EM41:   .ASCIZ/ERROR: UBE FAILED TO INT. ON DONE/
 3536  024256  020072  041125  020105
 3537  024264  040506  046111  042105
 3538  024272  052040  020117  047111
 3539  024300  027124  047440  020116
```

```
 3540  024306  047504  042516     000
 3541  024313     105  051122  051117  EM42:   .ASCIZ/ERROR: CCOVF NOT CLEARED BY GO/
 3542  024320  020072  041503  053117
 3543  024326  020106  047516  020124
 3544  024334  046103  040505  042522
 3545  024342  020104  054502  043440
 3546  024350  000117
 3547  024352  051105  047522  035122  EM43:   .ASCIZ/ERROR: UBE DID NOT DO DATO TO PROPER # OF LOCS. (4)/
 3548  024360  052440  042502  042040
 3549  024366  042111  047040  052117
 3550  024374  042040  020117  040504
 3551  024402  047524  052040  020117
 3552  024410  051120  050117  051105
 3553  024416  021440  047440  020106
 3554  024424  047514  051503  020056
 3555  024432  032050  000051
 3556  024436  042101  020104  051106  DH43:   .ASCIZ*ADD FROM/TO WERE WRITTEN*
 3557  024444  046517  052057  020117
 3558  024452  042527  042522  053440
 3559  024460  044522  052124  047105
 3560  024466     000
 3561  024467     105  051122  051117  EM44:   .ASCIZ/ERROR: INT. ON DONE BIT NOT CLEARED/
 3562  024474  020072  047111  027124
 3563  024502  047440  020116  047504
 3564  024510  042516  041040  052111
 3565  024516  047040  052117  041440
 3566  024524  042514  051101  042105
 3567  024532     000
 3568  024533     105  051122  051117  EM45:   .ASCIZ/ERROR: CCOVF NOT SET/
 3569  024540  020072  041503  053117
 3570  024546  020106  047516  020124
 3571  024554  042523  000124
 3572  024560  051105  047522  035122  EM46:   .ASCIZ/ERROR: DATO FROM BECC NOT DONE PROPERLY/
 3573  024566  042040  052101  020117
 3574  024574  051106  046517  041040
 3575  024602  041505  020103  047516
 3576  024610  020124  047504  042516
 3577  024616  050040  047522  042520
 3578  024624  046122  000131
 3579  024630  042101  020104  020040  DH46:   .ASCIZ*ADD    /DATA    /DATA SHOULD BE*
 3580  024636  042057  052101  020101
 3581  024644  020040  042057  052101
 3582  024652  020101  044123  052517
 3583  024660  042114  041040  000105
 3584  024666  051105  047522  035122  EM47:   .ASCIZ/ERROR: UBE DID NOT DO 2 DATO FOR EACH REQUEST/
 3585  024674  052440  042502  042040
 3586  024702  042111  047040  052117
 3587  024710  042040  020117  020062
 3588  024716  040504  047524  043040
 3589  024724  051117  042440  041501
 3590  024732  020110  042522  052521
 3591  024740  051505  000124
 3592  024744  051105  047522  035122  EM49:   .ASCIZ/ERROR: UBE DID NOT DO 2 DATIP FOR EACH REQUEST/
 3593  024752  052440  042502  042040
 3594  024760  042111  047040  052117
 3595  024766  042040  020117  020062
```

```
  3596  024774  040504  044524  020120
  3597  025002  047506  020122  040505
  3598  025010  044103  051040  050505
  3599  025016  042525  052123     000
  3600  025023     105  051122  051117  EM51:   .ASCIZ/ERROR: NPR DATO NOT DONE/
  3601  025030  020072  050116  020122
  3602  025036  040504  047524  047040
  3603  025044  052117  042040  047117
  3604  025052  000105
  3605  025054  051105  047522  035122  EM52:   .ASCIZ/ERROR: NPR DID NOT SET RDY/
  3606  025062  047040  051120  042040
  3607  025070  042111  047040  052117
  3608  025076  051440  052105  051040
  3609  025104  054504     000
  3610  025107     105  051122  051117  EM53:   .ASCIZ/ERROR: UBE DID NOT INT. WHEN NPR FINISHED/
  3611  025114  020072  041125  020105
  3612  025122  044504  020104  047516
  3613  025130  020124  047111  027124
  3614  025136  053440  042510  020116
  3615  025144  050116  020122  044506
  3616  025152  044516  044123  042105
  3617  025160     000
  3618  025161     105  051122  051117  EM54:   .ASCIZ/ERROR: TWO LOC. WRITTEN WHEN ONE NPR AND INT. ON DONE ENABLED/
  3619  025166  020072  053524  020117
  3620  025174  047514  027103  053440
  3621  025202  044522  052124  047105
  3622  025210  053440  042510  020116
  3623  025216  047117  020105  050116
  3624  025224  020122  047101  020104
  3625  025232  047111  027124  047440
  3626  025240  020116  047504  042516
  3627  025246  042440  040516  046102
  3628  025254  042105     000
  3629  025257     015  052012  051505  MSG3:   .ASCII<15><12>/TESTING UBE WILL NOT INTERRUPT DURING NPR/<15><12>
  3630  025264  044524  043516  052440
  3631  025272  042502  053440  046111
  3632  025300  020114  047516  020124
  3633  025306  047111  042524  051122
  3634  025314  050125  020124  052504
  3635  025322  044522  043516  047040
  3636  025330  051120  005015
  3637  025334  043111  042040  042517          .ASCIZ*IF DOES, CPU WILL GO DOWN*<15><12>*ENTERING TEST*
  3638  025342  026123  041440  052520
  3639  025350  053440  046111  020114
  3640  025356  047507  042040  053517
  3641  025364  006516  042412  052116
  3642  025372  051105  047111  020107
  3643  025400  042524  052123     000
  3644  025405     015  042412  044530  MSG4:   .ASCIZ<15><12>/EXITING TEST/
  3645  025412  044524  043516  052040
  3646  025420  051505  000124
  3647  025424  051105  047522  035122  EM56:   .ASCIZ/ERROR: TEST OF WRONG A  LINES ERROR BIT FAILED/
  3648  025432  052040  051505  020124
  3649  025440  043117  053440  047522
  3650  025446  043516  040440  020040
  3651  025454  044514  042516  020123
```

```
 3652  025462  051105  047522  020122
 3653  025470  044502  020124  040506
 3654  025476  046111  042105     000
 3655  025503     102  041505  031122  EM57:   .ASCIZ/BECR2 BIT 9 FALSELY SET/
 3656  025510  041040  052111  034440
 3657  025516  043040  046101  042523
 3658  025524  054514  051440  052105
 3659  025532     000
 3660  025533     124  020117  047111  EM58:   .ASCIZ/TO INTERRUPT CPU/
 3661  025540  042524  051122  050125
 3662  025546  020124  050103  000125
 3663  025554  051105  047522  035122  EM59:   .ASCIZ/ERROR: TEST OF NSSYN ERROR BIT FAILED/
 3664  025562  052040  051505  020124
 3665  025570  043117  047040  051523
 3666  025576  047131  042440  051122
 3667  025604  051117  041040  052111
 3668  025612  043040  044501  042514
 3669  025620  000104
 3670  025622  047524  051440  052105  EM60:   .ASCIZ/TO SET BIT 8 OF BECR2/
 3671  025630  041040  052111  034040
 3672  025636  047440  020106  042502
 3673  025644  051103  000062
 3674  025650  047524  051440  052105  EM61:   .ASCIZ/TO SET BIT 15 OF BECR1/
 3675  025656  041040  052111  030440
 3676  025664  020065  043117  041040
 3677  025672  041505  030522     000
 3678  025677     124  020117  046103  EM62:   .ASCIZ/TO CLEAR BIT 8 OF BECR2/
 3679  025704  040505  020122  044502
 3680  025712  020124  020070  043117
 3681  025720  041040  041505  031122
 3682  025726     000
 3683  025727     106  046101  042523  EM63:   .ASCIZ/FALSELY INTERRUPTED CPU/
 3684  025734  054514  044440  052116
 3685  025742  051105  052522  052120
 3686  025750  042105  041440  052520
 3687  025756     000
 3688  025757     105  051122  051117  EM64:   .ASCIZ/ERROR: TEST OF WRONG GRANT OR NOT ONE GRANT ERROR BITS FAILED/
 3689  025764  020072  042524  052123
 3690  025772  047440  020106  051127
 3691  026000  047117  020107  051107
 3692  026006  047101  020124  051117
 3693  026014  047040  052117  047440
 3694  026022  042516  043440  040522
 3695  026030  052116  042440  051122
 3696  026036  051117  041040  052111
 3697  026044  020123  040506  046111
 3698  026052  042105     000
 3699  026055     116  020117  051107  EM65:   .ASCIZ/NO GRANT OR NOT ONE GRANT ERROR BIT FALSELY SET/
 3700  026062  047101  020124  051117
 3701  026070  047040  052117  047440
 3702  026076  042516  043440  040522
 3703  026104  052116  042440  051122
 3704  026112  051117  041040  052111
 3705  026120  043040  046101  042523
 3706  026126  054514  051440  052105
 3707  026134     000
```

```
 3708  026135      040  044527  044124  DH65:   .ASCIZ/ WITH BECR1 = /
 3709  026142  041040  041505  030522
 3710  026150  036440  000040
 3711  026154  044527  044124  044440  EM66:   .ASCIZ/WITH INT. ON DONE = 1/
 3712  026162  052116  020056  047117
 3713  026170  042040  047117  020105
 3714  026176  020075  000061
 3715  026202  051127  047117  020107  EM67:   .ASCIZ/WRONG GRANT ERROR BIT FALSELY SET/
 3716  026210  051107  047101  020124
 3717  026216  051105  047522  020122
 3718  026224  044502  020124  040506
 3719  026232  051514  046105  020131
 3720  026240  042523  000124
 3721  026244  047524  041440  042514  EM69:   .ASCIZ/TO CLEAR BIT 11 OF BECR1/
 3722  026252  051101  041040  052111
 3723  026260  030440  020061  043117
 3724  026266  041040  041505  030522
 3725  026274      000
 3726  026275      105  051122  051117  EM70:   .ASCIZ/ERROR: TEST OF TIME DELAY AND BUS LATENCY FAILED/
 3727  026302  020072  042524  052123
 3728  026310  047440  020106  044524
 3729  026316  042515  042040  046105
 3730  026324  054501  040440  042116
 3731  026332  041040  051525  046040
 3732  026340  052101  047105  054503
 3733  026346  043040  044501  042514
 3734  026354  000104
 3735  026356  047524  051440  052105  EM71:   .ASCIZ/TO SET BIT 6 OF BECR2/
 3736  026364  041040  052111  033040
 3737  026372  047440  020106  042502
 3738  026400  051103  000062
 3739  026404  047524  041440  042514  EM72:   .ASCIZ/TO CLEAR BIT 6 OF BECR2/
 3740  026412  051101  041040  052111
 3741  026420  033040  047440  020106
 3742  026426  042502  051103  000062
 3743  026434  051105  047522  035122  EM73:   .ASCIZ/ERROR: TEST OF POWER DOWN BIT FAILED/
 3744  026442  052040  051505  020124
 3745  026450  043117  050040  053517
 3746  026456  051105  042040  053517
 3747  026464  020116  044502  020124
 3748  026472  040506  046111  042105
 3749  026500      000
 3750  026501      124  020117  047520  EM74:   .ASCIZ/TO POWER DOWN CPU/
 3751  026506  042527  020122  047504
 3752  026514  047127  041440  052520
 3753  026522      000
 3754  026523      124  020117  047520  EM75:   .ASCIZ/TO POWER UP CPU/
 3755  026530  042527  020122  050125
 3756  026536  041440  052520      000
 3757  026543      124  020117  042522  EM76:   .ASCIZ/TO RE POWER DOWN CPU/
 3758  026550  050040  053517  051105
 3759  026556  042040  053517  020116
 3760  026564  050103  000125
 3761  026570  047524  051440  052105  EM77:   .ASCIZ/TO SET BIT 4 OF BECR2/
 3762  026576  041040  052111  032040
 3763  026604  047440  020106  042502
```

```
 3764  026612  051103  000062
 3765  026616  051105  047522  035122  EM78:   .ASCIZ/ERROR: DCLO FAILED TO CLEAR REG/
 3766  026624  042040  046103  020117
 3767  026632  040506  046111  042105
 3768  026640  052040  020117  046103
 3769  026646  040505  020122  042522
 3770  026654  000107
 3771  026656  051105  047522  035122  EM80:   .ASCIZ/ERROR: DYNAMIC TEST OF UBE FAILED/
 3772  026664  042040  047131  046501
 3773  026672  041511  052040  051505
 3774  026700  020124  043117  052440
 3775  026706  042502  043040  044501
 3776  026714  042514  000104
 3777  026720  047524  046040  040517  EM81:   .ASCIZ/TO LOAD PROPER DATA/
 3778  026726  020104  051120  050117
 3779  026734  051105  042040  052101
 3780  026742  000101
 3781  026744  051105  047522  035122  EM82:   .ASCIZ/ERROR: TEST OF PASSING GRANTS FAILED/
 3782  026752  052040  051505  020124
 3783  026760  043117  050040  051501
 3784  026766  044523  043516  043440
 3785  026774  040522  052116  020123
 3786  027002  040506  046111  042105
 3787  027010     000
 3788  027011     105  051122  051117  EM83:   .ASCIZ/ERROR:FALSE INTERRUPT WHEN DO RELEASE BUS IMMED./
 3789  027016  043072  046101  042523
 3790  027024  044440  052116  051105
 3791  027032  052522  052120  053440
 3792  027040  042510  020116  047504
 3793  027046  051040  046105  040505
 3794  027054  042523  041040  051525
 3795  027062  044440  046515  042105
 3796  027070  000056
 3797  027072  051105  047522  035122  EM84:   .ASCIZ/ERROR:TEST OF MULTIPLE INTERRUPTS FAILED TO SET RDY/
 3798  027100  042524  052123  047440
 3799  027106  020106  052515  052114
 3800  027114  050111  042514  044440
 3801  027122  052116  051105  052522
 3802  027130  052120  020123  040506
 3803  027136  046111  042105  052040
 3804  027144  020117  042523  020124
 3805  027152  042122  000131
 3806  027156  041125  020105  044527  MSG7:   .ASCIZ/UBE WITH VECTOR: /
 3807  027164  044124  053040  041505
 3808  027172  047524  035122  000040
 3809  027200  040440  042116  041040  MSG8:   .ASCIZ/ AND BR AT: /
 3810  027206  020122  052101  020072
 3811  027214     000
 3812  027215     040  040506  051514  MSG9:   .ASCIZ/ FALSELY INTERRUPTED WHEN/<15><12>
 3813  027222  046105  020131  047111
 3814  027230  042524  051122  050125
 3815  027236  042524  020104  044127
 3816  027244  047105  005015     000
 3817  027251     040  044123  052517  MSG10:  .ASCIZ/ SHOULD HAVE INTERRUPTED/<15><12>
 3818  027256  042114  044040  053101
 3819  027264  020105  047111  042524
```

```
 3820  027272  051122  050125  042524
 3821  027300  006504  000012
 3822  027304  005015  040520  051523  MSG11:  .ASCIZ<15><12>/PASSING OF GRANTS NOT TESTED WITH ONE EXERCISOR/<15><12>
 3823  027312  047111  020107  043117
 3824  027320  043440  040522  052116
 3825  027326  020123  047516  020124
 3826  027334  042524  052123  042105
 3827  027342  053440  052111  020110
 3828  027350  047117  020105  054105
 3829  027356  051105  044503  047523
 3830  027364  006522  000012
 3831  027370  005015  043111  046440  MSG12:  .ASCII<15><12>/IF MORE THAN ONE UBE PRESENT JUMPER W1/<15><12>
 3832  027376  051117  020105  044124
 3833  027404  047101  047440  042516
 3834  027412  052440  042502  050040
 3835  027420  042522  042523  052116
 3836  027426  045040  046525  042520
 3837  027434  020122  030527  005015
 3838  027442  044123  052517  042114          .ASCIZ/SHOULD BE INSERTED IN ALL UBE EXCEPT LAST/<15><12>
 3839  027450  041040  020105  047111
 3840  027456  042523  052122  042105
 3841  027464  044440  020116  046101
 3842  027472  020114  041125  020105
 3843  027500  054105  042503  052120
 3844  027506  046040  051501  006524
 3845  027514  000012
 3846  027516  005015  042524  052123  MSG13:  .ASCIZ<15><12>/TESTING UBE WITH BEDB ADDRESS: /
 3847  027524  047111  020107  041125
 3848  027532  020105  044527  044124
 3849  027540  041040  042105  020102
 3850  027546  042101  051104  051505
 3851  027554  035123  000040
 3852  027560  005015  020040  047040  MSG14:  .ASCIZ<15><12>/   NOTE:DISREGARD BIT 13 =1 OF BECR2/<15><12>
 3853  027566  052117  035105  044504
 3854  027574  051123  043505  051101
 3855  027602  020104  044502  020124
 3856  027610  031461  036440  020061
 3857  027616  043117  041040  041505
 3858  027624  031122  005015     000
 3859  027631     015  050012  020103  MSG15:  .ASCIZ<15><12>/PC OF ERROR MESSAGE WAS: /
 3860  027636  043117  042440  051122
 3861  027644  051117  046440  051505
 3862  027652  040523  042507  053440
 3863  027660  051501  020072     000
 3864  027665     015  020012  020040  MSG16:  .ASCIZ<15><12>/     UNIBUS EXERCISER MODULE DIAGNOSTIC--CZKUB-B/<15><12><15><12>
 3865  027672  020040  047125  041111
 3866  027700  051525  042440  042530
 3867  027706  041522  051511  051105
 3868  027714  046440  042117  046125
 3869  027722  020105  044504  043501
 3870  027730  047516  052123  041511
 3871  027736  026455  055103  052513
 3872  027744  026502  006502  006412
 3873  027752  000012
 3874  027754  020040  020040  020040  MSG17:  .ASCIZ/         TEST NUMBER WAS: /
 3875  027762  020040  052040  051505
```

```
 3876  027770  020124  052516  041115
 3877  027776  051105  053440  051501
 3878  030004  020072     000
 3879
 3880          030010                          .EVEN
 3881                                  ;////////////////////////////////////////////////////////////////////////
 3882                                  ;BUFFER WORK AREA
 3883                                  ;////////////////////////////////////////////////////////////////////////
 3884  030010  000011                  BUFF1:  .BLKW   11
 3885          000001                          .END
```

```
A       003172          757#    771
ACALC   003376          792#    804     812     2352
B       001126          155     157#
BEBA    002522          662#    780*    794*    964     993     1290*   1307*   1325*   1344*   1393*   1420*   1452*   1484*
                        1513    1519*   1529*   1533    1535    1540    1586*   1592*   1599    1647*   1657    1684*   1748*
                        1788*   1839*   1886*   1898*   1961*   1981*   1984*   1985    2017*   2110*   2211*   2218*   2232
                        2233    2316*   2640*
BEBD    002516          660#    778*    792*    816     823     849     949     978     1012    1013    1295    1297    1308*
                        1314    1329    1333    1343*   1351    1401    1405    1424    1428    1584    1590    1597    1653
                        1683*   1790*   1885*   2642*
BECC    002520          661#    779*    793*    1289*   1306*   1324*   1345*   1392*   1421*   1454*   1486*   1510*   1521*
                        1530*   1581*   1587*   1594*   1626    1628    1648*   1655    1685*   1749*   1789*   1840*   1887*
                        1899*   1935*   1963*   1983*   2016*   2109*   2124*   2209*   2219*   2228    2230    2317*   2641*
BECR1   002524          663#    781*    795*    851     867     901     1020    1027    1031    1041    1106*   1107    1114
                        1116    1176*   1182*   1187*   1192*   1203*   1210    1212*   1213*   1232*   1261*   1292*   1310*
                        1327*   1346*   1362    1394*   1422*   1455*   1462    1479    1488*   1490    1511*   1522*   1531*
                        1582*   1588*   1595*   1619*   1633*   1651*   1687*   1705    1709*   1750*   1751    1791*   1841*
                        1888*   1900*   1913*   1936*   1939    1943*   1965*   1991*   2013*   2018*   2021    2031    2036*
                        2043    2049    2055    2060    2080*   2090    2111*   2126*   2213*   2220*   2238    2240    2248*
                        2285*   2318*   2639*   2663
BECR2   002526          664#    782*    796*    847*    858     860     869     871*    872     916     918     933     935
                        1029    1033    1039    1043    1076*   1077    1089*   1220    1222    1244    1260*   1269    1453*
                        1457    1469    1474    1485*   1520*   1524    1545    1593*   1650*   1664*   1707    1710    1962*
                        1967    1975    1982*   1987    1989*   1990*   1993    2023    2025    2029    2038    2040    2078*
                        2085    2102    2125*   2154*   2190    2197*   2210*   2222*   2227*   2235    2236    2247*   2249
                        2254*   2259*   2260    2262    2263    2638*
BEGO    002534          667#    784*    825     1592    1961    2286*   2418*   2507*
BERE    002530          665#    783*    797*    848*    870*    902*    1473*   1496*   2101*   2113*   2637*
BE1BA   002542          670#
BE1BD   002536          668#    786     790     2421
BE1CC   002540          669#
BE1CR1  002544          671#    2410*   2422*   2556*   2565
BE1CR2  002546          672#
BE1RE   002550          673#
BE1VEC  002552          674#    2391*   2392
BE2BA   002560          677#
BE2BD   002554          675#    2380    2424
BE2CC   002556          676#
BE2CR1  002562          678#    2411*   2425*   2557*   2567
BE2CR2  002564          679#
BE2RE   002566          680#
BE2VEC  002570          681#    2394*   2395
BE3BA   002576          684#
BE3BD   002572          682#    2427
BE3CC   002574          683#
BE3CR1  002600          685#    2412    2414*   2428*   2558    2560*   2569    2571
BE3CR2  002602          686#
BE3RE   002604          687#
BE3VEC  002606          688#    2397    2399*   2400
BE4BA   002614          691#
BE4BD   002610          689#    2430
BE4CC   002612          690#
BE4CR1  002616          692#    2415    2417*   2431*   2561    2563*   2573    2575
BE4CR2  002620          693#
BE4RE   002622          694#
BE4VEC  002624          695#    2402    2404*   2405
```

```
BIT0   = 000001            124#
BIT00  = 000001            114#    124
BIT01  = 000002            113#    123
BIT02  = 000004            112#    122
BIT03  = 000010            111#    121
BIT04  = 000020            110#    120
BIT05  = 000040            109#    119
BIT06  = 000100            108#    118
BIT07  = 000200            107#    117
BIT08  = 000400            106#    116
BIT09  = 001000            105#    115    2752    2810
BIT1   = 000002            123#
BIT10  = 002000            104#   2795
BIT11  = 004000            103#   2759
BIT12  = 010000            102#
BIT13  = 020000            101#   2802
BIT14  = 040000            100#   2734
BIT15  = 100000             99#
BIT2   = 000004            122#
BIT3   = 000010            121#
BIT4   = 000020            120#
BIT5   = 000040            119#
BIT6   = 000100            118#
BIT7   = 000200            117#
BIT8   = 000400            116#
BIT9   = 001000            115#
BPTVEC= 000014             131#
BUFF1    030010           1287*   1290    1298    1299    1305*   1307    1312    1315    1316    1323*   1325    1331    1334
                          1335    1341*   1344    1348    1352    1353    1391*   1393    1403    1406    1407    1419*   1420
                          1426    1429    1430    1529    1533    1536    1647    1649*   1657    1679    1681    1684    1696
                          1699    1703    1717    1722    1744    1746    1748    1753    1758    1784    1786    1788    1795
                          1798    1802    1807    1812    1835    1837    1839    1845    1848    1852    1857    1862    1884*
                          1886    1893    1895*   1896*   1898    1906    2017    2110    2311    2313    2316    2320*   2329
                          2333    2407    2439*   2448    2451    2501    2502    2515    2517    2519    2520    2522    2523
                          3884#
C        003500             802     806#
CLRREG   016170            1011    1038    1102    1159    1259    1273    1288    1304    1322    1389    1417    1450    1506
                          1562    1616    1646    1678    1742    1783    1834    1883    1930    1960    2012    2077    2107
                          2121    2283    2308    2351   2637#
CR     = 000015             39#   2996    3006
CRDY     016244           1395    1423    1456    1489    1512    1523    1532    1583    1589    1596    1665    1818    1868
                          1890    1902    1915    1966    1992    2020    2112    2127   2660#
CRLF   = 000200             40#   2967    3006
D        003514            807     809#
DB     = 170000            139#    749
DDISP  = 177570             46#    196     728
DH15     022133            309     524     549    3332#
DH17     022307            319    3352#
DH18     022423            324    3366#
DH19     022517            329    3377#
DH2      021020            244    3220#
DH24     023023            354     359     364     369     394     399    3413#
DH3      021121            249    3233#
DH34     023565            404    3479#
DH35     023661            409     419     424    3490#
DH36     023756            414    3501#
```

```
DH4     021202            254     259     264     624    3244#
DH43    024436            449     469     484    3556#
DH46    024630            464     639    3579#
DH65    026135            559     569     574    2466    3708#
DH7     021432            269    3273#
DINT    016274           1507    1563    1980    2108    2122    2212    2673#
DISPLA  001174            196#    728*    736*   2773*   2794*
DISPRE  000174            147#    736
DSWR  = 177570             45#    195     727
DT2     021046            245     320     410     420     425     450     470     485    3225#
DT24    023100            355     360     365     370     395     400    3422#
DT3     021130            250     310     325     330     405     415     525     550     560     570     575    3236#
DT4     021242            255     260     265     465     625     640    3251#
DT7     021510            270    3282#
E       003532            810     813#
EMAP    002510            657#    743*    752*    762     770     791
EMTVEC= 000030            134#    711*    712*
EM1     020676            238    3205#
EM10    021644            283    3298#
EM11    021705            288    3304#
EM12    021751            293    3311#
EM14    022016            303    3318#
EM15    022047            308    3323#
EM16    022155            313    3336#
EM17    022250            318    3346#
EM18    022334            323    3356#
EM19    022436            328    3368#
EM2     020760            243    3214#
EM20    022540            333    3380#
EM21    022607            338    3388#
EM22    022645            343    3394#
EM23    022710            348    3400#
EM24    022754            353    3406#
EM25    023112            358    3424#
EM26    023152            363    3430#
EM27    023213            368    3436#
EM28    023253            373    3442#
EM29    023302            378    3446#
EM3     021054            248    3226#
EM30    023331            383    3450#
EM31    023361            388    3455#
EM32    023411            393    3460#
EM33    023460            398    3467#
EM34    023515            403    3472#
EM35    023604            408    3482#
EM36    023707            413    3494#
EM37    023766            418    3503#
EM38    024025            423    3509#
EM39    024115            428    3519#
EM4     021134            253    3237#
EM40    024173            433    3527#
EM41    024251            438    3535#
EM42    024313            443    3541#
EM43    024352            448     508    3547#
EM44    024467            453    3561#
EM45    024533            458    3568#
```

```
EM46    024560             463    3572#
EM47    024666             468     473    3584#
EM49    024744             478     483    3592#
EM5     021252             258    3253#
EM51    025023             488    3600#
EM52    025054             493    3605#
EM53    025107             498    3610#
EM54    025161             503    3618#
EM56    025424             513    3647#
EM57    025503             518    3655#
EM58    025533             523    3660#
EM59    025554             528    3663#
EM6     021320             263    3260#
EM60    025622             533    3670#
EM61    025650             538    3674#
EM62    025677             543    3678#
EM63    025727             548    3683#
EM64    025757             553    3688#
EM65    026055             558    3699#
EM66    026154             563    3711#
EM67    026202             568    3715#
EM69    026244             578    3721#
EM7     021370             268    3267#
EM70    026275             583    3726#
EM71    026356             588    3735#
EM72    026404             593    3739#
EM73    026434             598    3743#
EM74    026501             603    3750#
EM75    026523             608    3754#
EM76    026543             613    3757#
EM77    026570             618    3761#
EM78    026616             623    3765#
EM8     021516             273    3283#
EM80    026656             633    3771#
EM81    026720             638    3777#
EM82    026744             643    3781#
EM83    027011             648    3788#
EM84    027072             653    3797#
EM9     021603             278    3292#
ERRVEC= 000004             127#    725     726*    737*    2739    2740*   2742*   2745*
ERR1    006620            1291    1370#
ERR2    006630            1309    1373#
ERR3    006640            1326    1376#
ERR4    006650            1342    1379#
F       003570             824#    826
FINT1   005172            1103    1126#
FINT3   005600            1173    1216#
FIRST   003632             829     830     843#
GNS   = ****** U           146    3114    3115    3116    3117    3118
HT    = 000011              37#   2965    3006
IADD    003266             765     778#
INTVEC  002532             666#    785*    798*   1103*   1104    1173*   1195*   1227*   1451*   1487*   1618*   1686*   1897*
                          1964*   2014*   2035*   2079*   2284*   2315*   2648    2650*   2673    2675*
IOTVEC= 000020             132#    709*    710*
ITRAP   004770            1069    1083#
LAST    014436             805     808    2354#
```

```
LAST1   014460            2354    2355    2380#
LF    = 000012              38#   3000    3006
LOOP1   003150             751#    756
LOOP2   003156             753#    769
MSG1    020342             761    3166#
MSG10   027251            2469    2554    3817#
MSG11   027304            2384    3822#
MSG12   027370             742    3831#
MSG13   027516             815    3846#
MSG14   027560             862     920    3852#
MSG15   027631            2710    3859#
MSG16   027665             741    3864#
MSG17   027754            2713    3874#
MSG2    020353            2589    3168#
MSG3    025257            1933    3629#
MSG4    025405            1946    3644#
MSG5    020611            1086    3196#
MSG7    027156            2462    2541    2548    3806#
MSG8    027200            2544    2551    3809#
MSG9    027215            2547    3812#
MTRAP   003234             747     767#
NO      002630             697#    788     790*    822     827*
NUBE    014426             922     959     988    1026    1121    1133    2351#
NUBE1   014432             884    2352#
PIRQ  = 177772              44#
PIRQVE= 000240             138#
PR0   = 000000              61#
PR1   = 000040              62#
PR2   = 000100              63#
PR3   = 000140              64#
PR4   = 000200              65#
PR5   = 000240              66#
PR6   = 000300              67#
PR7   = 000340              68#
PS    = 177776              41#     42
PSW   = 177776              42#    844*    898*    948*    977*   1010*   1073*   1101*   1109*   1158*   1177*   1181*   1186*
                           1196*   1202*   1231*   1258*   1262*   1286*   1293*   1390*   1418*   1449*   1467*   1508*   1564*
                           1615*   1620*   1632*   1634*   1645*   1652*   1677*   1688*   1743*   1782*   1792*   1794*   1816*
                           1833*   1842*   1844*   1866*   1882*   1901*   1914*   1937*   1959*   1972*   2015*   2019*   2076*
                           2095*   2123*   2148*   2214*   2246*   2251*   2282*   2287*   2310*   2409*   2419*   2436*   2455*
                           2457*   2486*   2489*   2508*   2564*
PTRAP   004754            1074    1080#
PWRVEC= 000024             133#    715*    716*   3125*   3126*   3135*   3141*   3153*   3154*
RCATCH  016222            1118    1132    1134    1234    1495    1550    1603    1635    1732    1919    1998    2063    2114
                          2132    2267    2291    2345   2647#
RDYS    006570            1294    1311    1328    1347   1361#
RESVEC= 000010             128#
RSTART  001100             152#
RTR     006044            1270   1273#
RVEC    016314             156    1205    1217    1240    1243   2681#
SPTR    002514             659#    745*    806     809     811*    813*
STACK = 001100              32#    707     744     843     897     947     976    1009    1061    1100    1157    1228    1257
                           1285   1388    1416    1448    1505    1561   1614    1644    1676    1741    1781    1832    1881
                           1929   1958    2011    2075    2120    2149   2281    2390
START   002632             150     153     157     700#    746    2703
START1  003100             740     743#   2629
```

```
STKLMT= 177774            43#
STRAP   004030           845    878#
SWR     001172           195#   705    727*   729    735*   739    759    809    1064   1083   1931   1944   2146
                        2206   2382   2459   2539   2708   2734   2752   2759   2795   2802   2807   2810   3133
                        3146*
SWREG   000176           148#   735
SW0    = 000001           96#
SW00   = 000001           86#    96
SW01   = 000002           85#    95
SW02   = 000004           84#    94
SW03   = 000010           83#    93
SW04   = 000020           82#    92
SW05   = 000040           81#    91    1064
SW06   = 000100           80#    90
SW07   = 000200           79#    89
SW08   = 000400           78#    88
SW09   = 001000           77#    87
SW1    = 000002           95#
SW10   = 002000           76#
SW11   = 004000           75#
SW12   = 010000           74#   739    759    1083   1931   1944   2382
SW13   = 020000           73#  2459   2539   2708
SW14   = 040000           72#
SW15   = 100000           71#
SW2    = 000004           94#
SW3    = 000010           93#
SW4    = 000020           92#
SW5    = 000040           91#
SW6    = 000100           90#
SW7    = 000200           89#
SW8    = 000400           88#
SW9    = 001000           87#
TBITVE= 000014           129#
TERRPC  016430           773    863    880    921    958    987   1025   1079   1120   1123   1127   1131   1207
                        1219   1224   1236   1242   1247   1272   1302   1319   1338   1356   1371   1374   1377
                        1380   1399   1410   1433   1461   1466   1472   1478   1483   1494   1538   1543   1549
                        1602   1631   1660   1663   1694   1713   1720   1725   1728   1731   1767   1810   1815
                        1860   1865   1904   1909   1912   1918   1942   1971   1978   1997   2033   2046   2052
                        2057   2062   2089   2094   2099   2106   2131   2158   2172   2185   2194   2244   2266
                        2290   2326   2344   2555   2708#
TKVEC  = 000060           136#
TMAP    002512           658#   791*   800*   801*   803
TPVEC  = 000064           137#
TRAPVE= 000034           135#   713*   714*
TRTVEC= 000014           130#
TSTA    006656          1303   1382#
TSTB    014132          2208   2271#
TST1    003630           842#
TST10   005224          1156#
TST11   005754          1225   1237   1245   1256#
TST12   006050          1284#
TST13   006656          1320   1339   1349   1357   1372   1375   1378   1387#
TST14   007022          1400   1404   1415#
TST15   007152          1427   1447#
TST16   007456          1504#
TST17   007740          1560#
```

```
TST2    004060            876     896#
TST20   010204           1613#
TST21   010332           1643#
TST22   010472           1675#
TST23   010776           1740#
TST24   011134           1760    1780#
TST25   011342           1831#
TST26   011550           1880#
TST27   012002           1927#
TST3    004266            946#
TST30   012116           1945    1957#
TST31   012372           2010#
TST32   012706           2074#
TST33   013134           2119#
TST34   013222           2144#
TST35   013476           2147    2204#
TST36   014132           2280#
TST37   014214           2306#
TST4    004364            975#
TST40   014456           2379#
TST5    004462           1008#
TST6    004656           1040    1060#
TST7    005032           1065    1099#
TYPDS = 104405           2618    3118#
TYPE  = 104401            741     742     761     764     815     818     862     920    1086    1933    1946    2384    2462
                         2466    2469    2541    2544    2547    2548    2551    2554    2589    2616    2619    2710    2713
                         2797    2805    2830    2847    2849    2852    2854    2858    2865    2927    2970    3066    3114#
                         3155
TYPOC = 104402            763     817    2464    2468    2543    2546    2550    2553    2712    2838    2862    3115#
TYPON = 104404           3117#
TYPOS = 104403           2715    3116#
T01L01  003676            850#    866
T01L02  003754            854     865#
T01L03  003712            855#    868     873
T01L04  003760            852     867#
T01L05  004042            864     882#
T01L06  003746            859     861     863#
T02L01  004214            911     923#
T02L02  004144            912#    930
T02L03  004122            904#    936
T02L04  004204            917     919     921#
T03L01  004346            953     960#
T03L02  004356            961     964#
T03L03  004312            951#    963
T03L04  004306            950#    965
T04L01  004444            982     989#
T04L02  004454            990     993#
T04L03  004410            980#    992
T04L04  004404            979#    994
T05L01  004572           1017    1019    1029#
T05L02  004624           1030    1038#
T05L03  004522           1016#   1035    1037
T05L04  004506           1013#   1044    1046
T05L05  004652           1042    1045#
T05L06  004620           1032    1036#
T05L07  004562           1021    1027#
```

```
T05L08  004540            1022#   1028
T06L01  005012            1082    1084    1087#
T07L03  005116            1111#   1115
T07L04  005162            1113    1122#
T07L05  005220            1117    1134#
T07L06  005210            1124    1132#
T07L07  005114            1110#   1128
T07L08  005202            1108    1130#
T08L01  005304            1169#   1172
T08L02  005552            1195    1209#
T08L03  005510            1199#   1214
T08L04  005642            1221    1227#
T08L05  005650            1227    1228#
T08L06  005736            1208    1244#
T08L07  005732            1211    1243#
T08L08  005252            1162#   1164
T08L09  005536            1198    1205#
T09L01  014210            2284    2291#
T10L01  006206            1296    1304#
T10L02  006326            1313    1322#
T10L03  006452            1332    1341#
T10L04  006616            1363    1369#
T10L05  006572            1362#   1366
T10L06  006412            1330    1333#
T11L01  006744            1397    1401#
T11L02  006764            1402    1405#
T12L01  007114            1425    1428#
T13L01  010012            1568    1575#
T13L02  007774            1570#   1573
T13L03  010016            1574    1577#
T13L04  010154            1585    1591    1599#
T13L05  010176            1598    1603#
T14L01  007670            1514    1540#
T14L02  007504            1510#   1518
T14L03  007712            1525    1545#
T14L04  007734            1534    1539    1544    1550#
T15L01  010250            1618    1622#
T15L02  010326            1625    1635#
T15L03  010234            1619#   1627
T16L01  010414            1653#   1654
T16L02  010452            1656    1662#
T16L03  010460            1658    1661    1664#
T17L01  010516            1680#   1682
T17L02  010612            1686    1696#
T17L03  010572            1690#   1692
T17L04  010710            1698    1715#
T17L05  010616            1697#   1700
T17L06  010630            1701#   1704
T17L07  010754            1706    1727#
T17L08  010764            1708    1730#
T17L09  010772            1695    1711    1714    1721    1726    1729    1732#
T17L10  010712            1702    1716#
T17L11  010732            1718    1722#
T18L01  011020            1745#   1747
T18L02  011052            1751#   1752
T18L03  011110            1756    1762#
```

```
T18L04  011072          1755#   1759
T19L01  011160          1785#   1787
T19L02  011270          1801    1806#
T19L03  011240          1796#   1799
T19L04  011252          1800#   1803
T19L05  011330          1804    1811    1816#
T19L07  011310          1808    1812#
T19L09  011266          1797    1805#
T20L01  011366          1836#   1838
T20L02  011474          1847    1855#
T20L03  011442          1846#   1849
T20L04  011456          1850#   1853
T20L05  011536          1854    1861    1866#
T20L06  011516          1858    1862#
T20L08  011476          1851    1856#
T21L01  011742          1892    1911#
T21L02  011770          1894    1917#
T21L03  011724          1897    1906#
T21L04  011776          1905    1907    1910    1916    1919#
T22L01  012076          1940    1943#
T23L01  007320          1451    1473#
T23L02  007254          1458    1462#
T23L03  007274          1463    1467#
T23L04  007446          1491    1495#
T23L05  007344          1475    1479#
T23L06  007364          1480    1484#
T23L07  007426          1487    1490#
T24L01  012230          1964    1975#
T24L02  012220          1968    1972#
T24L03  012366          1979    1988    1998#
T24L04  012334          1986    1991#
T24L05  012266          1983#   1994
T24L06  012250          1974    1980#
T25L01  012512          2014    2028#
T25L02  012542          2022    2035#
T25L03  012644          2024    2054#
T25L04  012664          2026    2059#
T25L05  012420          2015#   2027
T25L06  012560          2035    2038#
T25L07  012622          2039    2048#
T25L08  012702          2034    2041    2047    2053    2058    2063#
T26L01  013042          2079    2101#
T26L02  012752          2082#   2084
T26L03  013002          2086    2090#
T26L04  013022          2091    2095#
T26L05  013066          2100    2103    2107#
T27L01  013322          2152    2160#
T27L02  013442          2159    2173    2186    2188    2190#
T27L03  013366          2161    2174#
T27L04  013336          2164#   2166    2169
T27L05  013430          2174    2187#
T27L06  013400          2177#   2179    2182
T27L07  013462          2191    2195#
T28L01  013634          2221    2224#
T28L02  013750          2231    2234    2237    2242#
T28L03  013766          2241    2246#
```

```
T28L04  014116          2245    2261    2267#
T28L05  013646          2225    2227#
T28L06  014010          2249#   2250
T28L08  014044          2253    2256#
T28L09  014056          2257    2259#
T28L10  013522          2207    2209#
T29L01  014242          2312#   2314
T29L02  014334          2315    2328#
T29L03  014372          2331    2339#
T29L04  014236          2311#   2337
T29L05  014342          2330#   2334
T29L06  014306          2320#   2323
T29L07  014420          2327    2338    2345#
T30L01  014514          2381    2390#
T30L02  014726          2391    2421#
T30L03  014742          2394    2424#
T30L04  014756          2399    2427#
T30L05  014772          2404    2430#
T30L06  015004          2423    2426    2429    2432#
T30L07  014714          2413    2416    2418#
T30L08  015504          2497    2505    2525#
T30L09  015330          2497#   2500
T30L10  015420          2506    2510#
T30L11  015406          2507#   2514
T30L12  015674          2470    2516    2540    2556#
T30L13  015472          2518    2522#
T30L14  015356          2503#   2521    2524
T30L15  015550          2530    2533#
T30L16  015570          2534    2537#
T30L17  015576          2532    2536    2538#
T30L20  016020          2582#   2586
T30L21  014634          2398    2403    2407#
T30L22  015024          2435    2438#
T30L25  015200          2452    2471#
T30L26  015076          2454#   2474    2484
T30L27  015246          2476    2481#
T30L28  015042          2448#   2492
T30L29  015116          2458#   2477
T30L30  015070          2453#   2479
T31L01  013216          2129    2132#
UCNT    002626           696#    814*   2434    2445
UINT    005716          1169    1239#
$AUTOB  001166           192#
$BDADR  001154           187#
$BDDAT  001160           189#
$BELL   001240           217#   2797    2822
$CHARC  017662          2972*   2982*   2989    2998*   3003#
$CKSWR= ****** U        3121
$CMTAG  001132           175#    702     703     711     717     718     719
$CM1  = 000004           207#    208#    209#    210#    211#
$CM2  = 000010           207#    208#    209#    210#    211#
$CM3  = 000004           205#    207
$CM4  = 000004           211#    212#    213#    214#    215#
$CRLF   001245           219#    764     818    2805    2822    2830    2849    2854    2858    2971    3006
$DBLK   017436          2893    2927    2935#
$DOAGN  016144          2612    2621    2627#
```

```
$DTBL   017426            2896    2931#
$ENDAD  016134             164    2623#   2817
$ENDCT  016102             717    2614#
$ENDMG  016153            2616    2631#
$ENULL  016150            2619    2630#
$EOP    016046             774    2386    2588    2604#
$EOPCT  016074             717*   2611#   2615
$ERFLG  001135             178#   2725    2748    2750    2756*   2777    2792*   2822
$ERMAX  001147             184#    720*   2750    2772*   2777
$ERROR  016720             711    2791#
$ERRPC  001150             185#   2711    2799*   2800*   2801    2822    2836
$ERRTB  001250             236#   2844
$ERRTY  017066            2804    2829#
$ERTTL  001144             182#   2798*   2822
$ESCAP  001236             216#    719*   2771*   2813    2815    2822
$FILLC  001210             203#   2975    3006
$FILLS  001207             202#   3006
$GDADR  001152             186#
$GDDAT  001156             188#
$GET42  016124            2620#
$GTSWR= ****** U          3120
$HD   = 000001              11      12
$ICNT   001136             179#   2763*   2764    2766*   2776
$ILLUP  020324            3125    3141    3158#
$INTAG  001167             193#
$ITEMB  001146             183#   2801*   2822    2833
$LF     001246             220#   2822    3006
$LPADR  001140             180#    721*    829*   2354*   2754*   2769*   2774    2776
$LPERR  001142             181#    722*    746*    830*   2355*   2754    2770*   2776    2812
$MAIL = ****** U           739    2769    2807    2959
$MXCNT  016716            2767    2776#
$NULL   001206             201#   2977    3006
$NWTST= 000001             835#    837     886#    888     938#    940     967#    969     996#    998    1048#   1050   1091#
                          1093    1136#   1138    1249#   1251    1275#   1277    1384#   1412#   1436#   1438    1499#  1501
                          1552#   1554    1606#   1608    1637#   1639    1667#   1669    1734#   1736    1769#   1771   1820#
                          1822    1870#   1872    1921#   1923    1949#   1951    2000#   2002    2065#   2067    2116#  2134#
                          2136    2199#   2201    2273#   2275    2294#   2296    2360#   2362
$OCNT   020110            3038*   3067*   3080#
$OMODE  020112            3033*   3037*   3042    3045*   3056*   3082#
$OVER   016702            2735    2755    2765    2773#
$PASS   001132             176#   2587    2608*   2609*   2617    2630    2761    2777
$POWER  020332            3156    3161#
$PWRDN  020164             715    3125#   3153
$PWRMG  020320            3156#
$PWRUP  020236            3135    3141#
$QUES   001244             218#   2822    3006
$RDCHR= ****** U          3121
$RDDEC= ****** U          3121
$RDLIN= ****** U          3121
$RDOCT= ****** U          3121
$REGAD  001212             205#
$REGO   001214             207#    855*    878*    912*    954*    983*   1022*   1174*   1179*   1184*   1193*   1199*  1222*
                          1230*   1297*   1314*   1333*   1351*   1405*   1428*   1469*   1535*   1540*   1547*   1599*  1600*
                          1628*   1722*   1763*   1812*   1862*   1975*   2029*   2031*   2043*   2049*   2055*   2060*  2228*
                          2232*   2235*   2238*   2242    2262*   2339*   2461*   2463    2526*   2542    3225    3236   3251
                          3282    3422
```

```
$REG1   001216          208#   856*   913*   956*   985*  1023*  1175*  1180*  1185*  1194*  1200*  1229*  1231
                       1298*  1315*  1334*  1352*  1406*  1429*  1536*  1541*  1629*  1723*  1764*  1813*  1863*
                       2242*  2263*  2340*  2465*  2467   2527*  2545   3225   3251   3282   3422
$REG2   001220          209#   914*   955*   984*  1299*  1316*  1335*  1353*  1407*  1430*  1765*  2229*  2239*
                       2264*  2341*  2528*  2549   3251   3422
$REG3   001222          210#  1300*  1317*  1336*  1354*  1408*  1431*  2531*  2535*  2537*  2552   3422
$RTNAD  016146         2629#
$R2A  = ****** U       3121
$SAVRE= ****** U       3121
$SAVR6  020330         3134*  3142   3143*  3144*  3160#
$SCOPE  016470          709   2733#
$SETUP= 000037          140#   708    709    711    713    715    717    718    719    721   2606   2734   2792
                       2810   2817
$STUP = 177777          140#
$SVLAD  016654         2743   2768#
$SVPC = 001132          162#   167
$SWR  = 167000            1#    11     16     17     18     19     20     21     22     23    215    216    217
                        718    719    721    722    843    897    947    976   1009   1061   1100   1157   1257
                       1285   1388   1416   1448   1505   1561   1614   1644   1676   1741   1781   1832   1881
                       1928   1958   2011   2075   2120   2145   2205   2281   2307   2380   2601   2607   2622
                       2628   2630   2726   2727   2728   2729   2730   2734   2746   2748   2749   2750   2757
                       2758   2759   2770   2773   2776   2783   2784   2785   2786   2787   2795   2802   2807
                       2810   2822   3157
$SWRMK= 000000         2730
$TIMES  001234          215#   718*  1085*  1928*  2145*  2205*  2307*  2385*  2607*  2757*  2764   2767*  2776
$TKB    001200          198#
$TKS    001176          197#
$TMP0   001224          211#  1160*  2681
$TMP1   001226          212#   903*   904    926    934*
$TMP2   001230          213#
$TMP3   001232          214#
$TN   = 000041           11#   835    843#   876    886    897#   938    947#   967    976#   996   1009#  1040
                       1048   1061#  1091   1100#  1136   1157#  1225   1237   1245   1249   1257#  1275   1285#
                       1320   1339   1349   1357   1372   1375   1378   1384   1388#  1400   1404   1412   1416#
                       1427   1436   1448#  1499   1505#  1552   1561#  1606   1614#  1637   1644#  1667   1676#
                       1734   1741#  1760   1769   1781#  1820   1832#  1870   1881#  1921   1928#  1945   1949
                       1958#  2000   2011#  2065   2075#  2116   2120#  2134   2145#  2147   2199   2205#  2273
                       2281#  2294   2307#  2360   2380#
$TPB    001204          200#  2995*  3006
$TPFLG  001211          204#  2953   3006
$TPS    001202          199#  2993   3006
$TRAP   020114          713   3091#
$TRAP2  020136         3102#  3113
$TRP  = 000006         3106#  3115#  3116#  3117#  3118#  3119#
$TRPAD  020150         3096   3113#
$TSTNM  001134          177#   831*  2356*  2606*  2714   2725   2768*  2773   2777   2794   2822
$TYPBN= ****** U       3119
$TYPDS  017222         2881#  3118
$TYPE   017446         2953#  3106   3114
$TYPEC  017616         2974   2981   2988   2993#  2994
$TYPEX  017664         2999   3001   3004#
$TYPOC  017712         3036#  3115
$TYPON  017726         3035   3038#  3117
$TYPOS  017666         3031#  3116
$XTSTR  016500         2737#
$$GET4= 000000         2622#
```

```
$OFILL  020111          3032*   3036*   3046    3081#
$40CAT= ****** U        2734    2804
.     = 030032           142#    146#    151#    162     163#    165#    167#    174#    221     706     721     722     2630
                        2634    2776    2777    2822    2868#   2935#   3006    3137    3159    3250#   3281#   3880#   3884#
```

```
COMMEN    139#
ENDCOM    139#
ERROR      33#   772   857   879   915   957   986  1024  1078  1119  1122  1126  1130  1206  1218
          1223  1235  1241  1246  1271  1301  1318  1337  1355  1370  1373  1376  1379  1398  1409
          1432  1459  1460  1464  1465  1470  1471  1476  1477  1481  1482  1492  1493  1537  1542
          1548  1601  1630  1659  1662  1693  1712  1719  1724  1727  1730  1766  1809  1814  1859
          1864  1903  1908  1911  1917  1941  1969  1970  1976  1977  1995  1996  2028  2030  2032
          2042  2044  2045  2048  2050  2051  2054  2056  2059  2061  2087  2088  2092  2093  2097
          2098  2104  2105  2130  2156  2157  2170  2171  2183  2184  2192  2193  2243  2265  2289
          2324  2325  2342  2343  2458  2538
ESCAPE    139#
GETPRI    139#
GETSWR    139#
MSG       835#   837   886#   888   938#   940   967#   969   996#   998  1048#  1050  1091#  1093  1136#
          1138  1249#  1251  1275#  1277  1436#  1438  1499#  1501  1552#  1554  1606#  1608  1637#  1639
          1667#  1669  1734#  1736  1769#  1771  1820#  1822  1870#  1872  1921#  1923  1949#  1951  2000#
          2002  2065#  2067  2134#  2136  2199#  2201  2273#  2275  2294#  2296  2360#  2362
MULT      139#
NEWTST    139#   835   886   938   967   996  1048  1091  1136  1249  1275  1384  1412  1436  1499
          1552  1606  1637  1667  1734  1769  1820  1870  1921  1949  2000  2065  2116  2134  2199
          2273  2294  2360
POP       139#  2922  3146  3147
PUSH      139#  2881  3127  3133
REPORT    139#
SCOPE      34#   842   896   946   975  1008  1060  1099  1156  1256  1284  1387  1415  1447  1504
          1560  1613  1643  1675  1740  1780  1831  1880  1927  1957  2010  2074  2119  2144  2204
          2280  2306  2350  2379  2605
SETPRI    139#
SETTRA   3106#  3115  3116  3117  3118
SETUP     139#   700
SKIP      139#   876  1040  1208  1211  1225  1237  1245  1320  1339  1349  1357  1372  1375  1378
          1400  1404  1427  1534  1539  1544  1695  1711  1714  1721  1726  1729  1760  1804  1811
          1854  1861  1907  1945  2147
SLASH     139#   775   777   819   821  1063  1068  2347  2349  2634  2636  2644  2646  2657  2659
          2670  2672  2678  2680  2705  2707  3881  3883
SPACE     139#
STARS     139#   160   170   221   698   835   841   886   895   938   945   967   974   996  1007
          1048  1059  1091  1098  1136  1155  1249  1255  1275  1283  1384  1386  1412  1414  1436
          1446  1499  1503  1552  1559  1606  1612  1637  1642  1667  1674  1734  1739  1769  1779
          1820  1830  1870  1879  1921  1926  1949  1956  2000  2009  2065  2073  2116  2118  2134
          2143  2199  2203  2273  2279  2294  2305  2360  2378  2598  2722  2779  2824  2871  2938
          3008  3085  3123  3139  3164
SWRSU     139#   723#
TRMTRP   3106#
TYPBIN    139#
TYPDEC    139#  2617
TYPNAM    139#
TYPNUM    139#
TYPOCS    139#
TYPOCT    139#   762   816  2463  2467  2542  2545  2549  2552  2711  2836  2860
TYPTXT    139#
$$CMRE    168#   207   208   209   210
$$CMTM    168#   211   212   213   214
$$ESCA    139#
$$NEWT    139#   835   886   938   967   996  1048  1091  1136  1249  1275  1384  1412  1436  1499
          1552  1606  1637  1667  1734  1769  1820  1870  1921  1949  2000  2065  2116  2134  2199
```

```
          2273    2294    2360
$$SET     3106#   3115    3116    3117    3118
$$SKIP     139#    876    1040    1225    1237    1245    1320    1339    1349    1357    1372    1375    1378    1400    1404
          1427    1760    1945    2147
.EQUAT       1#     29
.HEADE       1#
.SETUP       1#    140
.SWRHI       1#     12
.SWRLO      23#     27
.$ACT1       1#    158
.$CATC       1#    140
.$CMTA       1#    168
.$EOP        1#   2596
.$ERRO       1#   2777
.$ERRT       1#   2822
.$POWE       1#   3121
.$SCOP       1#   2720
.$TRAP       1#   3083
.$TYPD       1#   2869
.$TYPE       1#   2936
.$TYPO       1#   3006


. ABS.  030032     000


 ERRORS DETECTED:  0

 CZKUBB.BIN,CZKUBB.LST/CRF/SOL/NL:TOC-CZKUBB.P11
 RUN-TIME: 22 14 1 SECONDS
 RUN-TIME RATIO: 524/38-13.4
 CORE USED:  20K  (39 PAGES)
```