PDP-11

UNIBUS SYS EXER

CZKUAE0

AH-8856E-MC

COPYRIGHT 75-80

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

Identification

Product Code:	AC-8855E-MC
Product Name:	CZKUAE0 Unibus Systems Exerciser Diagnostic
DATE:	NOV 79
Maintainer:	Diagnostic Group
Author:	Manuel Soares
MODIFIED BY:	BILL SCHLITZKUS

Table of Contents

1.0 ABSTRACT

2.0 REQUIREMENTS

2.1 Hardware
2.2 Software

3.0 PROGRAM DESCRIPTION

3.1 Switch Options
3.2 Test 1 thru Test 16
3.3 Sysmac Routines

4.0 ERROR REPORTING

1.0 ABSTRACT

This program was created to test PDP-11's CPU interface circuitry. It uses the Unibus Exerciser(s) (UBE) to insure proper operation by simulating peripherals which would require the 11-CPU to produce the necessary signals. It should be noted that the UBE is a powerful tool and if it is not programmed correctly could cause various problems on the Bus.

2.0 REQUIREMENTS

2.1 Hardware

This program assumes the following in proper working condition: 1. The Unibus, 2. Memory (8K minimum), and 3. UBE(s) (4 maximum). If a fourth UBE is being used, its time delay should be set at 100us to prevent latency problems in one of the tests.

With two or more UBE(s), all should have W1 jumpers except the one furthest electrically from the CPU. If there are more than 4 UBE(s) on the Unibus the program is not responsible for any problems which might occur , since it is programmed to handle a maximum of only 4.

2.2 Software

After loading the program the starting address must be 200, so that the first time through, the available UBE(s) are determined. In addition if one or more UBE(s) are added or removed, the program again must be started at 200. Otherwise, to avoid duplicating some printouts, the program can be restarted at address 220.

```
****************************************************************
A SOFTWARE HALT CAN BE CAUSED BY DEPRESSING CONTROL-H ON THE CONSOLE.
IF THE PROGRAM IS HALTED THIS WAY, AND THE PROGRAM IS RESTARTED,
DEPRESS ANY CONSOLE KEY TO REMOVE THE SOFTWARE HALT CONDITION.
****************************************************************
```

3.0 PROGRAM DESCRIPTION

This program was assembled with MACY11 using PDP-11 Maindec Sysmac package .

3.1 Switch Options

The use of thes program on processors having a software switch register necessitates operator interaction: the operator must set up location 176 with the switch register values desired .

```
     Switch        Use
     ------        ---

       15      Halt on Error
       14      Loop on test
       13      Inhibit error typeouts
       11      Inhibit iterations
       10      Bell on error
        9      Loop on error
        8      Loop on test in SWR<5:0>
       NOTE:   If you wish to inhibit all typing except ''end of pass''
        you must put down switch 7, after loading 200.
        6        WHEN SET, INHIBIT TEST 14
```

3.2 Test 1 through Test 16

TEST 1 - No Bus grants issued with processor at higher priority than bus request. This test is to insure that any request is not honored as long as the processor is at the same or higher priority.

TEST 2 - Issuing of non-processor grants and arbitration tests. This test will request on NPR through BR4 levels with the processor status initially at level 7 and make sure the device exercises an NPG to do a fun 1-dati, then the requests will be repeated while sequentially lowering the processor status from 7 to 0 to allow arbitration of all requests and the issuing of NPG.

TEST 3 - Issuing of Bus grant 7 and arbitration tests. This test will arbitrate for a BG7. The requests will be on levels BR7 thru BR4, doing fun 1-dati transfers, and the processor status lowered sequentially from 7 to 0.

TEST 4 - Issuing of Bus grant 6 and arbitration tests. This test will arbitrate for a BG6, the requests will be on levels BR6 thru BR4, doing fun 1-dati transfers, and the processor status lowered sequentially from 6 to 0.

TEST 5 - Issuing of Bus grant 5 and arbitration tests. This test will arbitrate for a BG5, the requests will be on levels BR5 thru BR4, doing fun 1-dati transfers, and the processor status lowered sequentially from 5 to 0.

```
TEST  6 - Issuing of Bus grant 4 and  arbitration  tests.   This  test
          will arbitrate for a BG4, the requests will be on level BR4,
          doing  func  1-dati  transfers,  and  the  processor  status
          lowered sequentially from 4 to 0.

TEST  7 - CPU test for no sack time out.  This test  will  check  that
          the  CPU  times  out  and drops a grant if no sack signal is
          received.  If the CPU  time  out  is  inoperative,  the  Bus
          exerciser  will time out and send the sack signal to prevent
          a Bus hang and set an error flag in CR2.

TEST 10 - CPU test for receiving sack.  This test is  to  insure  that
          the CPU can receive the sack signal; The time delay will be
          set on device 1 and several dati transfers made.   If  there
          is  not bus late error, the CPU received sack correctly.  It
          is assumed that dev 1 time delay is set for 10us.

TEST 11 - Passing of grants and interrupt test.  This  test  will  set
          off   all   available   devices  simultaneously  whose  only
          functions will be to interrupt, the requests will all be  at
          level 7 so that the device closest to tha CPU should receive
          BG7 first and  interrupt  first,  the  next  closest  should
          interrupt next and so on.

TEST 12 - Address lines (14 - 17) check.  This  test  will  check  Bus
          address  lines 14 thru 17 by doing a fun 1-dati-npr to those
          addresses.  If  the  addresses  don't  exist  the  interrupt
          routine will ignore any no ssyn error.

TEST 13 - CPU test for  ACLO/DCLO  sequence.   This  test  checks  the
          assertion  of  ACLO  and  DCLO and that the CPU traps to the
          correct service routine.  If this program is  running  under
          ACT11 this test will be skipped.

TEST 14 - Parity error test.  This test will cause  parity  error  and
          checks  that  the CPU traps to the correct vector.
        THIS TEST IS SKIPPED ON MACHINES THAT DON'T HAVE THE SXT INSTRUCTION
        (EG., 11/05 AND 11/20).
        THIS TEST SHOULD BE DESELECTED IF THE MEMORY PARITY OPTION
        IS NOT PRESENT OR NOT ENABLED.

        SW06=1          INHIBIT TEST 14

TEST 15 - Multitransfers I.  This test will cause any Bus  exercisers,
          up  to  4,  to  create a lot of traffic on the Bus and check
          that the CPU  can  handle  it;   all  devices  are  set  off
          simultaneously.

TEST 16 - Multitransfers  II.  This  test  will  have  the   available
          exercisers  doing  various  transfers  and/or  interrupts at
          different request  levels  to  further  check  CPU  handling
          capabilities.
```

TEST 17 - DUMMY END OF PROGRAM. This portion of the program is just to see if ''^H'' has been typed on the console to cause a program halt. If there is no ''^H'' the program continues by jumping to $EOP (end-of-pass routine).
IF THE PROGRAM IS HALTED THIS WAY, AND THE PROGRAM IS RESTARTED, DEPRESS ANY CONSOLE KEY TO REMOVE THE SOFTWARE HALT CONDITION.

3.3 Sysmac Routines

The 'END OF PASS ROUTINE' thru 'Power Down and Up Routines', as listed in the program listing, are the Sysmac package macros. They are called out in the source program, some with arguments and some without, and are expanded in the listing. Some macros are necessary for the operation of others, so for a complete explanation of all available Sysmac Macros see PDP-11 Maindec Sysmac Package (DZQAC-B-D).

4.0 ERROR REPORTING

The minimum amount of information given when an error occurs is the PC of the error call and the Test number in which it occurred. Other pertinent data will by typed out depending on the test being run at that time.

```
  17        OPERATIONAL SWITCH SETTINGS
  29        BASIC DEFINITIONS
 139        MEMORY MANAGEMENT DEFINITIONS
 196        TRAP CATCHER
 226        ACT11 HOOKS
 237        COMMON TAGS
 295        ERROR POINTER TABLE
 528        INITIALIZE THE COMMON TAGS
 713        T1      NO BUS GRANTS ISSUED WITH PROCESSOR AT HIGHER PRIORITY THAN BUS REQUEST
 744        T2      ISSUING OF NON-PROCESSOR GRANTS AND ARBITRATION TESTS
 782        T3      ISSUING OF BUS GRANT 7 AND ARBITRATION TESTS
 818        T4      ISSUING OF BUS GRANT 6 AND ARBITRATION TESTS
 854        T5      ISSUING OF BUS GRANT 5 AND ARBITRATION TESTS
 890        T6      ISSUING OF BUS GRANT 4 AND ARBITRATION TESTS
 926        T7      CPU TEST FOR NO SACK TIME OUT
 966        T10     CPU TEST FOR RECEIVING SACK
1007        T11     PASSING OF GRANTS AND INTERRUPT TEST
1080        T12     ADDRESS LINES (14 - 17) CHECK
1123        T13     CPU TEST FOR ACLO/DCLO SEQUENCE
1165        T14     PARITY ERROR TEST
1217        T15     MULTITRANSFERS I
1280        T16     MULTITRANSFERS II
1406        T17     DUMMY END OF PROGRAM
1803        END OF PASS ROUTINE
1852        SCOPE HANDLER ROUTINE
1917        ERROR HANDLER ROUTINE
1962        ERROR MESSAGE TYPEOUT ROUTINE
2009        TTY INPUT ROUTINE
2083        ROUTINE TO SIZE MEMORY
2197        SAVE AND RESTORE R0-R5 ROUTINES
2242        TYPE ROUTINE
2331        BINARY TO OCTAL (ASCII) AND TYPE
2408        CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
2475        TRAP DECODER
2498        TRAP TABLE
2517        POWER DOWN AND UP ROUTINES
```

```
     1
```

```
       167400                  $SWR=167400
       000300                  $SWRMK=300
                       .TITLE  UNIBUS EXERCISER
                       ;*COPYRIGHT (C) SEPT 79
                       ;*DIGITAL EQUIPMENT CORP.
                       ;*MAYNARD, MASS. 01754
                       ;*
                       ;*PROGRAM BY DIAG. ENG.
                       ;*
                       ;*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
                       ;*        11            INHIBIT ITERATIONS
                       ;*        10            BELL ON ERROR
                       ;*         9            LOOP ON ERROR
                       ;*         8            LOOP ON TEST IN SWR<5:0>
                       ;*          6           WHEN SET, INHIBIT TEST 14
                       .SBTTL  BASIC DEFINITIONS

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

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

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

```

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

                                ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
        100000                  SW15=   100000
        040000                  SW14=   40000
        020000                  SW13=   20000
        010000                  SW12=   10000
        004000                  SW11=   4000
        002000                  SW10=   2000
        001000                  SW09=   1000
        000400                  SW08=   400
        000200                  SW07=   200
        000100                  SW06=   100
        000040                  SW05=   40
        000020                  SW04=   20
        000010                  SW03=   10
        000004                  SW02=   4
        000002                  SW01=   2
        000001                  SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

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

```
                             .EQUIV  BIT08,BIT8
                             .EQUIV  BIT07,BIT7
                             .EQUIV  BIT06,BIT6
                             .EQUIV  BIT05,BIT5
                             .EQUIV  BIT04,BIT4
                             .EQUIV  BIT03,BIT3
                             .EQUIV  BIT02,BIT2
                             .EQUIV  BIT01,BIT1
                             .EQUIV  BIT00,BIT0

                             ;*BASIC "CPU" TRAP VECTOR ADDRESSES
        000004               ERRVEC= 4                ;;TIME OUT AND OTHER ERRORS
        000010               RESVEC= 10               ;;RESERVED AND ILLEGAL INSTRUCTIONS
        000014               TBITVEC=14               ;;"T" BIT
        000014               TRTVEC= 14               ;;TRACE TRAP
        000014               BPTVEC= 14               ;;BREAKPOINT TRAP (BPT)
        000020               IOTVEC= 20               ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
        000024               PWRVEC= 24               ;;POWER FAIL
        000030               EMTVEC= 30               ;;EMULATOR TRAP (EMT) **ERROR**
        000034               TRAPVEC=34               ;;"TRAP" TRAP
        000060               TKVEC=  60               ;;TTY KEYBOARD VECTOR
        000064               TPVEC=  64               ;;TTY PRINTER VECTOR
        000240               PIRQVEC=240              ;;PROGRAM INTERRUPT REQUEST VECTOR
                             .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                             ;*KT11 VECTOR ADDRESS

        000250               MMVEC=  250

                             ;*KT11 STATUS REGISTER ADDRESSES

        177572               SR0=     177572
        177574               SR1=     177574
        177576               SR2=     177576
        172516               SR3=     172516

                             ;*USER "I" PAGE DESCRIPTOR REGISTERS

        177600               UIPDR0= 177600
        177602               UIPDR1= 177602
        177604               UIPDR2= 177604
        177606               UIPDR3= 177606
        177610               UIPDR4= 177610
        177612               UIPDR5= 177612
        177614               UIPDR6= 177614
        177616               UIPDR7= 177616

                             ;*USER "I" PAGE ADDRESS REGISTERS

        177640               UIPAR0= 177640
        177642               UIPAR1= 177642
        177644               UIPAR2= 177644
        177646               UIPAR3= 177646
        177650               UIPAR4= 177650
        177652               UIPAR5= 177652
        177654               UIPAR6= 177654
        177656                  UIPAR7= 177656

                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

        172300                  KIPDR0= 172300
        172302                  KIPDR1= 172302
        172304                  KIPDR2= 172304
        172306                  KIPDR3= 172306
        172310                  KIPDR4= 172310
        172312                  KIPDR5= 172312
        172314                  KIPDR6= 172314
        172316                  KIPDR7= 172316

                                ;*KERNEL "I" PAGE ADDRESS REGISTERS

        172340                  KIPAR0= 172340
        172342                  KIPAR1= 172342
        172344                  KIPAR2= 172344
        172346                  KIPAR3= 172346
        172350                  KIPAR4= 172350
        172352                  KIPAR5= 172352
        172354                  KIPAR6= 172354
        172356                  KIPAR7= 172356

                                .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
        000200                  .=200
000200  005037  001176                  CLR     $TMP0            ;MAKE SURE TMP0=0
000204  000137  001760                  JMP     @#START          ;JUMP TO START


;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;*WHEN LOADING THE PROGRAM FOR THE FIRST TIME, OR ANY TIME
        ;*AFTER ALTERING THE # OF EXERCISERS ON THE BUS,
        ;*YOU MUST START AT LOCATION 200 AND
        ;*RESTART AT LOCATION 220 ONLY IF YOU DO NOT WISH
        ;*TO SIZE MEMORY AND TYPE OUT DEV ADDRESSES AGAIN
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

        000220                  .=220
000220  012737  000777  001176          MOV     #777,$TMP0       ;TMP0 IS INDICATOR FOR RESTART
000226  000137  001760                  JMP     @#START          ;JUMP TO START

                                .SBTTL  ACT11 HOOKS

```

```
                                 ;;*******************************************************************
                                 ;HOOKS REQUIRED BY ACT11
        000232                           $SVPC=.                        ;SAVE PC
        000046                           .=46
000046  015376                           $ENDAD                         ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                           .=52
000052  040000                           .WORD   40000                  ;;2)SET LOC.52 TO 40000
        000232                           .=$SVPC                        ;; RESTORE PC .
```

```
                                          .SBTTL  COMMON TAGS

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

              001100                              .=1100
      001100                              $CMTAG:                                 ;;START OF COMMON TAGS
      001100  000000                      $PASS:  .WORD   0                       ;;CONTAINS PASS COUNT
      001102     000                      $TSTNM: .BYTE   0                       ;;CONTAINS THE TEST NUMBER
      001103     000                      $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
      001104  000000                      $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
      001106  000000                      $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
      001110  000000                      $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
      001112  000000                      $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
      001114     000                      $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
      001115     001                      $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
      001116  000000                      $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
      001120  000000                      $GDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
      001122  000000                      $BDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
      001124  000000                      $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
      001126  000000                      $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
      001130  000000                              .WORD   0                       ;;RESERVED--NOT TO BE USED
      001132  000000                              .WORD   0
      001134     000                      $AUTOB: .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
      001135     000                      $INTAG: .BYTE   0                       ;;INTERRUPT MODE INDICATOR
      001136  000000                              .WORD   0
      001140  177570                      SWR:    .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
      001142  177570                      DISPLAY: .WORD  DDISP                   ;;ADDRESS OF DISPLAY REGISTER
      001144  177560                      $TKS:   177560                          ;;TTY KBD STATUS
      001146  177562                      $TKB:   177562                          ;;TTY KBD BUFFER
      001150  177564                      $TPS:   177564                          ;;TTY PRINTER STATUS REG. ADDRESS
      001152  177566                      $TPB:   177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
      001154     000                      $NULL:  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
      001155     002                      $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
      001156     012                      $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A ''LINE FEED''
      001157     000                      $TPFLG: .BYTE   0                       ;;''TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
      001160  000000                      $REGAD: .WORD   0                       ;;CONTAINS THE ADDRESS FROM
                                                                                  ;;WHICH  ($REG0) WAS OBTAINED
      001162  000000                      $REG0:  .WORD   0                       ;;CONTAINS (($REGAD)+0)
      001164  000000                      $REG1:  .WORD   0                       ;;CONTAINS (($REGAD)+2)
      001166  000000                      $REG2:  .WORD   0                       ;;CONTAINS (($REGAD)+4)
      001170  000000                      $REG3:  .WORD   0                       ;;CONTAINS (($REGAD)+6)
      001172  000000                      $REG4:  .WORD   0                       ;;CONTAINS (($REGAD)+10)
      001174  000000                      $REG5:  .WORD   0                       ;;CONTAINS (($REGAD)+12)
      001176  000000                      $TMP0:  .WORD   0                       ;;USER DEFINED
      001200  000000                      $TMP1:  .WORD   0                       ;;USER DEFINED
      001202  000000                      $TMP2:  .WORD   0                       ;;USER DEFINED
      001204  000000                      $TMP3:  .WORD   0                       ;;USER DEFINED
      001206  000000                      $TMP4:  .WORD   0                       ;;USER DEFINED
      001210  000000                      $TMP5:  .WORD   0                       ;;USER DEFINED
      001212  000000                      $TIMES: 0                               ;;MAX. NUMBER OF ITERATIONS
      001214  000000                      $ESCAPE:0                               ;;ESCAPE ON ERROR ADDRESS
      001216  177607  000377              $BELL:  .ASCIZ  <207><377><377>         ;;CODE FOR BELL
      001222     077                      $QUES:  .ASCII  /?/                     ;;QUESTION MARK
      001223     015                      $CRLF:  .ASCII  <15>                    ;;CARRIAGE RETURN
```

```
   291  001224  000012                  $LF:     .ASCIZ  <12>            ;;LINE FEED
   292                                  ;;******************************************************************
```

```
                                .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


001226                          $ERRTB:
                                ;;*******************************************************************
                                ;;*******************************************************************
                                ;ITEM 1
001226  011144                          EM1             ;CPU TRAPPED THRU LOC 4 -TIME OUT
001230  011212                          DH1             ; ADDR   $ERRPC #ERR/TST#
001232  015022                          DT1             ;$REG2,$ERRPC,$TSTNM,0
001234  000000                          0
                                ;ITEM 2
001236  011243                          EM2             ;CPU ISSUED A BUS GRANT WITH PSW = 7
                                                        ;DEV 1 SHOULD NOT HAVE BECOME BUS MASTER
001240  011356                          DH2             ;BE1DB BE1CC BE1BA BE1CR1 PSW $ERRPC #ERR/TST#
001242  015032                          DT2             ;$REG0,$REG1,$REG2,$REG3,$REG4,$ERRPC,$TSTNM,0
001244  000000                          0
                                ;ITEM  3
001246  011446                          EM3             ;CPU DID NOT ISSUE A BUS NPG
001250  011500                          DH3             ;BE1CR1 BE1CC FM-PS TO-PS $ERRPC #ERR/TST#
001252  015052                          DT3             ;$REG0,$REG1,$REG2,$REG3,$ERRPC,$TSTNM,0
001254  000000                          0
                                ;ITEM 4
001256  011561                          EM4             ;CPU DID NOT ISSUE BUS GRANT 7
001260  011500                          DH3
001262  015052                          DT3
001264  000000                          0
                                ;ITEM 5
001266  011617                          EM5             ;CPU DID NOT ISSUE BUS GRANT 6
001270  011500                          DH3
001272  015052                          DT3
001274  000000                          0
                                ;ITEM 6
001276  011655                          EM6             ;CPU DID NOT ISSUE BUS GRANT 5
001300  011500                          DH3
001302  015052                          DT3
001304  000000                          0
                                ;ITEM 7
001306  011713                          EM7             ;CPU DID NOT ISSUE BUS GRANT 4
001310  011500                          DH3
001312  015052                          DT3
001314  000000                          0
                                ;ITEM 10
001316  011751                          EM10            ;ONE OR MORE DEVICES DID NOT INTERRUPT
001320  012017                          DH10            ;THIS IS THE ORDER IN WHICH THEY INTERRUPTED
```

```
                                             ;  1ST  2ND  3RD  4TH $ERRPC #ERR/TST#
001322  015070                DT10           ;$REG1,$REG2,$REG3,$REG4,$ERRPC,$TSTNM,0
001324  000000                0
                        ;ITEM 11
001326  012155                EM11           ;BUS ADDRESS LINES <A17:A14> DID NOT FUNCTION PROPERLY
001330  012243                DH11           ;BE1CR1 BE1CR2 BE1BA $ERRPC #ERR/TST#
001332  015106                DT11           ;$REG1,$REG2,$REG3,$ERRPC,$TSTNM,0
001334  000000                0
                        ;ITEM 12
001336  012314                EM12           ;CPU NO SACK TIME OUT LOGIC FAILED(TO NEGATE BUS GRANT)
001340  012402                DH12           ;BE1CR1 BE1CR2 $ERRPC #ERR/TST#
001342  015122                DT12           ;$REG0,$REG1,$ERRPC,$TSTNM,0
001344  000000                0
                        ;ITEM 13
001346  012443                EM13           ;CPU DID NOT PROPERLY EXECUTE AN ACLO/DCLO SEQUENCE
001350  012526                DH13           ;$ERRPC #ERR/TST#
001352  015134                DT13           ;$ERRPC,$TSTNM,0
001354  000000                0
                        ;ITEM 14
001356  012547                EM14           ;CPU DID NOT TRAP FROM BUS PARITY ERR PA/PB = 0/1
001360  012526                DH13
001362  015134                DT13
001364  000000                0
                        ;ITEM 15
001366  012632                EM15           ;DEV 1 DID DATIP WITH ROL ON DATOB TO MEMORY
                                             ;THE TRANSFER TO THE FOLLOWING LOC WAS INCORRECT
001370  012775                DH15           ;MEMORY  ACTUAL  CORRECT
                                             ; LOC     DATA    DATA  $ERRPC #ERR/TST# $ICNT #
001372  015142                DT15           ;$REG0,$REG1,$REG3,$ERRPC,$TSTNM,$ICNT,0
001374  000000                0
                        ;ITEM 16
001376  013107                EM16           ;DEV 3'S DATO TO MEMORY DID NOT EQUAL PATTERN IN R3
001400  012775                DH15
001402  015142                DT15
001404  000000                0
                        ;ITEM 17
001406  013175                EM17           ;DEV 4'S DATO TO MEMORY DID NOT EQUAL PATERN IN R4
001410  012775                DH15
001412  015142                DT15
001414  000000                0
                        ;ITEM 20
001416  013263                EM20           ;DEV 1 DID FUN 1-NPR-DATIP;INCORRECT PATTERN IN MEMORY
001420  012775                DH15
001422  015142                DT15
001424  000000                0
                        ;ITEM 21
001426  013357                EM21           ;DEV 2 DID FUN 2-NPR-DATOB;INCORRECT PATTERN IN MEMORY
001430  012775                DH15
001432  015142                DT15
001434  000000                0
                        ;ITEM 22
001436  013453                EM22           ;BIT 7 OF CR2 SET-CPU DID NOT TIME OUT WITH SACK INHIBITED
001440  013545                DH22           ;DEV #  PC  $ERRPC #ERR/TST#
001442  015160                DT22           ;$TMP4,$REG5,$ERRPC,$TSTNM,0
001444  000000                0
                        ;ITEM 23
```

```
001446  013607                      EM23                  ;BIT 11 OF CR2 SET-NO SSYN ON INTR SIGNAL
001450  013545                      DH22
001452  015160                      DT22
001454  000000                      0
                            ;ITEM 24
001456  013660                      EM24                  ;BIT 5 OF CR2 SET-RECEIVED WRONG GRANT
001460  013545                      DH22
001462  015160                      DT22
001464  000000                      0
                            ;ITEM 25
001466  013726                      EM25                  ;BIT 6 OF CR2 SET-BUS LATE
001470  013545                      DH22
001472  015160                      DT22
001474  000000                      0
                            ;ITEM 26
001476  013760                      EM26                  ;BIT 8 OF CR2 SET-DEV DID NOT RECEIVE SSYN
001500  013545                      DH22
001502  015160                      DT22
001504  000000                      0
                            ;ITEM 27
001506  014022                      EM27                  ;BIT 9 OF CR2 SET-WRONG ADDR ON BUS
001510  013545                      DH22
001512  015160                      DT22
001514  000000                      0
                            ;ITEM 30
001516  014071                      EM30                  ;BIT 10 OF CR2 SET-DEV RECEIVED OTHER THAN ONE GRANT
001520  013545                      DH22
001522  015160                      DT22
001524  000000                      0
                            ;ITEM31
001526  014160                      EM31                  ;BKGRND RTN INSTRUCTIONS OF NEGB'S WERE NOT DONE
                                                          ;CORRECTLY TO $REG1 DURING MULTITRANFERS II
001530  014320                      DH31                  ;ACTUAL CORRECT
                                                          ;DATA    DATA $ERRPC #ERR/TST# $ICNT #
001532  015172                      DT31                  ;$REG1,146463,$ERRPC,$TSTNM,$ICNT,0
001534  000000                      0
                            ;ITEM 32
001536  014413                      EM32                  ;DEV 3 DID DATI BUT HAS INCORRECT
                                                          ;VALUES IN DATA REGISTER
001540  014320                      DH31
001542  015172                      DT31
001544  000000                      0
                            ;ITEM 33
001546  014477                      EM33                  ;DEV 4 DID NOT INTR THE CORRECT # OF TIMES
001550  014320                      DH31
001552  015172                      DT31
001554  000000                      0
                            ;ITEM 34
001556  014551                      EM34                  ;LAST DATI XFER BY DEV 1 WAS INCORRECT-
                                                          ;EITHER DEV DID NOT WORK OR WRONG DATA WASSET UP
001560  014320                      DH31
001562  015172                      DT31
001564  000000                      0
                            ;ITEM 35
001566  014725                      EM35                  ;CPU TRAPPED THRU LOC 0 TO CATCH
                                                          ;IMPROPERLY LOADED VECTORS
```

```
001570  011212                            DH1                 ; ADDR    $ERRPC #ERR/TST#
001572  015022                            DT1                 ;$REG2,$ERRPC,$TSTNM,0
001574  000000                            0
                                  ;;*************************************************************
                                  ;;*************************************************************
                                  ;;*************************************************************
001576  007740                            ALLERR   :7740              ;ALL ERR BITS OF CR2
001600  170014                            SIMLGO   :170014            ;ADDR TO SET OFF ALL DEVICES SIMOLTANEOUSLY
        000114                            PBVEC    =114               ;TRAP VEC FOR PARITY ERROR
        000116                            PBPSW    =116               ;PSW ADDR FOR TRAP ON PARITY ERR
001602  000000                            BE1DB    :0                 ;DATA REG ADDR FOR DEVICE 1
001604  000000                            BE1CC    :0                 ;CYCLE COUNT REG ADDR FOR DEV 1
001606  000000                            BE1BA    :0                 ;ADDR REG ADDR FOR DEV 1
001610  000000                            BE1CR1   :0                 ;CONTROL REG 1 ADDR FOR DEV 1
001612  000000                            BE1CLR   :0                 ;CLEAR ERRS REG ADDR FOR DEV 1
001614  000000                            BE1CR2   :0                 ;CONTROL REG 2 ADDR FOR DEV 1
001616  000000                            BE2DB    :0                 ;DATA REG ADDR FOR DEV 2
001620  000000                            BE2CC    :0                 ;CYCLE COUNT REG ADDR FOR DEV 2
001622  000000                            BE2BA    :0                 ;ADDR REG ADDR FOR DEV 2
001624  000000                            BE2CR1   :0                 ;CONTROL REG 1 ADDR FOR DEV 2
001626  000000                            BE2CLR   :0                 ;CLEAR ERRS REG ADDR FOR DEV 2
001630  000000                            BE2CR2   :0                 ;CONTROL REG 2 ADDR FOR DEV 2
001632  000000                            BE3DB    :0                 ;DATA REG ADDR FOR DEV 3
001634  000000                            BE3CC    :0                 ;CYCLE COUNT REG ADDR FOR DEV 3
001636  000000                            BE3BA    :0                 ;ADDR REG ADDR FOR DEV 3
001640  000000                            BE3CR1   :0                 ;CONTROL REG 1 ADDR FOR DEV 3
001642  000000                            BE3CLR   :0                 ;CLEAR ERRS REG ADDR FOR DEV 3
001644  000000                            BE3CR2   :0                 ;CONTROL REG 2 ADDR FOR DEV 3
001646  000000                            BE4DB    :0                 ;DATA REG ADDR FOR DEV 4
001650  000000                            BE4CC    :0                 ;CYCLE COUNT REG ADDR FOR DEV 4
001652  000000                            BE4BA    :0                 ;ADDR REG ADDR FOR DEV 4
001654  000000                            BE4CR1   :0                 ;CONTROL REG 1 ADDR FOR DEV 4
001656  000000                            BE4CLR   :0                 ;CLEAR ERRS REG ADDR FOR DEV 4
001660  000000                            BE4CR2   :0                 ;CONTROL REG 2 ADDR FOR DEV 4
001662  000000                            BE1VEC   :0                 ;TRAP VEC ADDR FOR DEV 1
001664  000000                            BE1PSW   :0                 ;PSW ADDR FOR TRAP THRU BE1VEC
001666  000000                            BE2VEC   :0                 ;TRAP VEC ADDR FOR DEV 2
001670  000000                            BE2PSW   :0                 ;PSW ADDR FOR TRAP THRU BE2VEC
001672  000000                            BE3VEC   :0                 ;TRAP VEC ADDR FOR DEV 3
001674  000000                            BE3PSW   :0                 ;PSW ADDR FOR TRAP THRU BE3VEC
001676  000000                            BE4VEC   :0                 ;TRAP VEC ADDR FOR DEV 4
001700  000000                            BE4PSW   :0                 ;PSW ADDR FOR TRAP THRU BE4VEC
001702  000000                            DEVCNT   :0                 ;CONTAINS # OF DEVICES ON BUS
001704  000000  000000  000000            DEVS     :0,0,0,0           ;WILL CONTAIN ADDR(S) OF INTR'G DEVS
001712  000000
001714  000000                            DATA1    :0                 ;MAX ADDR TO WHICH DATA XFERRED BY DEV 1
001716  000000                            DATA2    :0                 ;MAX ADDR TO WHICH DATA XFERRED BY DEV 2
001720  000000                            DATA3    :0                 ;MAX ADDR TO WHICH DATA XFERRED BY DEV 3
001722  000000                            DATA4    :0                 ;MAX ADDR TO WHICH DATA XFERRED BY DEV 4
001724  000000                            ENDMEM   :0                 ;TAG ENDING DEFINED LABELS
                                  ;;*************************************************************
                                  ;;*************************************************************
001726                            CLRRTN:
001726  012703  001602                    MOV      #BE1DB,R3          ;R3 IS POINTER TO BUFFER AREAS
001732  005023                    1$:     CLR      (R3)+              ;CLEAR BUFFER THEN INCREMENT ADDR
001734  022703  001724                    CMP      #ENDMEM, R3        ;IF POINTER AT LAST BUFFER, EXIT
```

```
001740  100374                           BPL     1$                 ;IF PLUS, GO BACK AND CLEAR NEXT ADDR
001742  012703  001162                   MOV     #$REG0,R3          ;NOW START TO CLEAR TEMP REGISTERS
001746  005023                   2$:     CLR     (R3)+              ;CLEAR CURRENT ADDR
001750  022703  001210                   CMP     #$TMP5,R3          ;CHECK FOR LAST TEMP REG ADDR
001754  101374                           BHI     2$                 ;IF NOT, CLEAR NEXT TEMP REG
001756  000207                           RTS     PC                 ;EXIT
                                 ;;*************************************************************************
                                 ;;*************************************************************************
001760                           START:
                                 .SBTTL  INITIALIZE THE COMMON TAGS
                                 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001760  012706  001100                   MOV     #$CMTAG,R6          ;;FIRST LOCATION TO BE CLEARED
001764  005026                           CLR     (R6)+               ;;CLEAR MEMORY LOCATION
001766  022706  001140                   CMP     #SWR,R6 ;;DONE?
001772  001374                           BNE     .-6                 ;;LOOP BACK IF NO
001774  012706  001100                   MOV     #STACK,SP           ;;SETUP THE STACK POINTER
                                 ;;INITIALIZE A FEW VECTORS
002000  012737  015416  000020           MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002006  012737  000340  000022           MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002014  012737  015674  000030           MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002022  012737  000340  000032           MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002030  012737  020252  000034           MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002036  012737  000340  000036           MOV     #340,@#TRAPVEC+2;LEVEL 7
002044  012737  020332  000024           MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002052  012737  000340  000026           MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002060  013737  015242  015234           MOV     $ENDCT,$EOPCT      ;;SETUP END-OF-PROGRAM COUNTER
002066  005037  001212                   CLR     $TIMES             ;;INITIALIZE NUMBER OF ITERATIONS
002072  005037  001214                   CLR     $ESCAPE            ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002076  112737  000001  001115           MOVB    #1,$ERMAX          ;;ALLOW ONE ERROR PER TEST
002104  012737  002104  001106           MOV     #.,$LPADR          ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002112  012737  002112  001110           MOV     #.,$LPERR          ;;SETUP THE ERROR LOOP ADDRESS
                                 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                 ;;EQUAL TO A ''-1'', SETUP FOR A SOFTWARE SWITCH REGISTER.
002120  013746  000004                   MOV     @#ERRVEC,-(SP)     ;;SAVE ERROR VECTOR
002124  012737  002160  000004           MOV     #64$,@#ERRVEC      ;;SET UP ERROR VECTOR
002132  012737  177570  001140           MOV     #DSWR,SWR          ;;SETUP FOR A HARDWARE SWICH REGISTER
002140  012737  177570  001142           MOV     #DDISP,DISPLAY     ;;AND A HARDWARE DISPLAY REGISTER
002146  022777  177777  176764           CMP     #-1,@SWR           ;;TRY TO REFERENCE HARDWARE SWR
002154  001012                           BNE     66$                ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                    ;;AND  THE HARDWARE SWR IS NOT = -1
002156  000403                           BR      65$                ;;BRANCH IF NO TIMEOUT
002160  012716  002166           64$:    MOV     #65$,(SP)          ;;SET UP FOR TRAP RETURN
002164  000002                           RTI
002166  012737  000176  001140   65$:    MOV     #SWREG,SWR         ;;POINT TO SOFTWARE SWR
002174  012737  000174  001142           MOV     #DISPREG,DISPLAY
002202  012637  000004           66$:    MOV     (SP)+,@#ERRVEC     ;;RESTORE ERROR VECTOR

002206  032777  000200  176724           BIT     #BIT07,@SWR        ;IS SWITCH 7 UP?
002214  001402                           BEQ     3$                 ;IF NOT, SKIP TYPEOUT
002216  104401  010754                   TYPE    ,QNO
002222                           3$:
002222  022737  000777  001176           CMP     #777,$TMP0         ;IS THIS RESTART FROM LOC 220?
002230  001002                           BNE     5$                 ;IF NOT,SKIP THE JMP INSTR
002232  000137  003252                   JMP     @#TST1             ;ELSE JUMP TO TEST 1

002236                           5$:
 002236  012737  010342  000000          MOV     #THRU0,0             ;SET UP FOR TRAP THRU LOC 0
 002244  012737  000340  000002          MOV     #PR7,2               ;SET UP PSW FOR TRAP THRU 0
 002252  032777  000200  176660          BIT     #BIT07,aSWR          ;IS SWITCH 7 UP?
 002260  001400                          BEQ     33$                  ;IF NOT, SKIP TYPEOUT
 002262                          33$:
 002262  004737  001726                  JSR     PC,CLRRTN            ;CLEAR BUFFER AREAS
 002266  012737  000340  000006          MOV     #PR7,ERRVEC+2        ;PS=7 FOR TRAP THRU LOC 4
 002274  012700  170000                  MOV     #170000,R0           ;SET UP POINTER FOR 1ST POSSIBLE DEV ADDR
 002300  012702  000510                  MOV     #510,R2              ;SET UP POINTER FOR 1ST POSSIBLE VEC ADDR
 002304  012701  001602                  MOV     #BE1DB,R1            ;SET UP POINTER FOR DEVICE ADDR LOCATION
 002310  012703  001662                  MOV     #BE1VEC,R3           ;SET UP POINTER FOR INTR ADDR LOCATION
 002314                          LODDEV:
 002314  022700  170060                  CMP     #170060,R0           ;IS R0 > LAST POSSIBLE DEV ADDR?
 002320  002002                          BGE     10$                  ;IF NOT,GO TO 10$
 002322  000137  002624                  JMP     BGIN                 ;ELSE GO TO BGIN
 002326                          10$:
 002326  012737  002432  000004          MOV     #NODEV,ERRVEC        ;SET UP TRAP VECTOR FOR TIME OUT
 002334  005710                          TST     (R0)                 ;SEE IF ACTUAL DEVICE ADDRESS EXISTS
 002336  012737  002550  000004          MOV     #TYMOUT,ERRVEC       ;CHANGE TRAP VECTOR FOR ERROR CONDITION
 002344  005237  001702                  INC     DEVCNT               ;COUNT DEVICES
 002350  010021                  MOVREG: MOV     R0,(R1)+             ;MOVE ACTUAL DEVICE ADDR TO DEVICE NAME
 002352  010037  001174                  MOV     R0,$REG5             ;REG5 CONTAINS LAST DEVICE ADDR
 002356  062700  000002                  ADD     #2,R0                ;INCREMENT POINTER BY 2
 002362  105237  001176                  INCB    $TMP0                ;COUNT # OF REGISTERS PER DEVICE
 002366  122737  000005  001176          CMPB    #5,$TMP0             ;AFTER 5 REGISTERS
 002374  001365                          BNE     MOVREG               ;ARE RECORDED
 002376  105037  001176                  CLRB    $TMP0                ;CLEAR THE COUNTING REGISTER
 002402  062700  000004                  ADD     #4,R0                ;ADD 4 TO THE POINTER THEN
 002406  010021                          MOV     R0,(R1)+             ;RECORD THE LAST REGISTER ADDRESS
 002410  062700  000002                  ADD     #2,R0                ;INCREMENT POINTER BY 2
 002414                          MOVVEC:                              ;NOW START RECORDING
 002414  010223                          MOV     R2,(R3)+             ;THE INTR VECTORS
 002416  062702  000002                  ADD     #2,R2                ;INCREMENT POINTER BY 2
 002422  010223                          MOV     R2,(R3)+             ;THE INTR VECTORS
 002424  062702  000002                  ADD     #2,R2                ;INCREMENT POINTER BY 2
 002430  000731                          BR      LODDEV               ;AND GO SEE IF THER'S ANOTHER DEVICE
                                 ;;*******************************************************************
                                 ;;*******************************************************************
 002432                          NODEV:                               ;TIME OUT ROUTINE FOR DEVICE CHECK
 002432  022700  170060                  CMP     #170060,R0           ;IF ALL POSSIBLE ADDR'S HAVE NOT BEEN CHECKED
 002436  003035                          BGT     ADD20                ;OUT-GO BACK AND CHECK FOR MORE,
 002440  012716  002624                  MOV     #BGIN,(SP)           ;ELSE CHANGE STACK POINTER
 002444  022737  000000  001702          CMP     #0,DEVCNT            ;CHECK FOR NO EXERCISERS
 002452  001035                          BNE     EXNO                 ;IF ONE OR MORE EXERCISERS, EXIT
 002454  104401  002462                  TYPE    ,65$                 ;;TYPE ASCIZ STRING
 002460  000423                          BR      64$                  ;;GET OVER THE ASCIZ
                                 ;;65$:  .ASCIZ  <15><12>/THERE ARE NO EXERCISERS ON THE BUS/
 002530                          64$:
 002530  000000                          HALT
 002532  062700  000020          ADD20:  ADD     #20,R0               ;ADD 20 TO POINTER
 002536  062702  000004                  ADD     #4,R2                ;POINTER=NEXT DEV'S VEC LOCATIONS
 002542  012716  002314                  MOV     #LODDEV,(SP)         ;GO BACK TO LODDEV
 002546                          EXNO:
 002546  000002                          RTI                          ;EXIT
                                 ;;*******************************************************************
                                 ;;*******************************************************************
```

```
002550                                  TYMOUT:                                         ;TIME OUT ROUTINE

002550  011637  001166                          MOV     (SP),$REG2              ;THE MOVE IS FOR TYPEOUT REASONS
002554  162737  000002  001166                  SUB     #2,$REG2                ;SUBTRACT 2 TO FIND ACTUAL ADDR
002562  104001                                  ERROR   1                       ;ERR MESSG FOR ILLEGAL TIME OUT
002564  000002                                  RTI
                                        ;;*******************************************************************
                                        ;;*******                R2                      ;ADD 1 TO R2(DEVICE COUNTER)
002614  020237  001702                          CMP     R2,DEVCNT               ;SEE IF IT = PREVIOUS COUNT
002620  101766                                  BLOS    1$                      ;IF NOT, CLEAR NEXT DEV REGS
002622  000207                                  RTS     PC                      ;EXIT
                                        ;//////////////////////////////////////////////////////////////////////
                                        ;//////////////////////////////////////////////////////////////////////
                                        ;//////////////////////////////////////////////////////////////////////
002624                                  BGIN:
002624  012737  010122  000024                  MOV     #PWRFAL,PWRVEC          ;TAKE CARE OF BIT 4(S) BEING SET RANDOMLY IN CR2(S)
002632  004737  010020                          JSR     PC,STVEC                ;SET UP VEC(S) FOR RANDOM ERRS
002636  004737  002566                          JSR     PC,CLRREG               ;CLEAR CONTENTS OF ALL AVAILABLE DEVS
002642  005037  001176                          CLR     $TMP0                   ;CLEAR TEMPORARY REG
002646  005037  001162                          CLR     $REG0                   ;CLEAR COUNTER
002652  032777  000200  176260                  BIT     #BIT07,@SWR             ;IS SWITCH 7 UP?
002660  001002                                  BNE     2$                      ;IF UP, GO TO 2$
002662  000137  003252                          JMP     5$                      ;ELSE SKIP THE TYPEOUTS
002666                                  2$:
002666  104401  002674                          TYPE    ,65$                    ;;TYPE ASCIZ STRING
002672  000431                                  BR      64$                     ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/THE FOLLOWING # OF EXERCISERS ARE ON THE BUS:  /
002756                                  64$:
002756  013746  001702                          MOV     DEVCNT,-(SP)            ;;SAVE DEVCNT FOR TYPEOUT
002762  104403                                  TYPOS                           ;;GO TYPE--OCTAL ASCII
002764     001                                  .BYTE   1                       ;;TYPE 1 DIGIT(S)
002765     000                                  .BYTE   0                       ;;SUPPRESS LEADING ZEROS
002766  104401  002774                          TYPE    ,67$                    ;;TYPE ASCIZ STRING
002772  000436                                  BR      66$                     ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>/THE LOWEST ELECT. PRIORITY UBE SHOULD NOT HAVE W1 JUMPER/
003070                                  66$:
003070  104401  003076                          TYPE    ,69$                    ;;TYPE ASCIZ STRING
003074  000415                                  BR      68$                     ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <15><12>/DEVICE ADDRESS(ES): /<15><12>
003130                                  68$:
003130  005037  001176                          CLR     $TMP0                   ;CLEAR TMP0(USED AS COUNTER)
003134  012700  001602                          MOV     #BE1DB,R0               ;USE R0 AS POINTER TO ADDRESSES
003140                                  4$:
003140  005237  001176                          INC     $TMP0                   ;ADD 1 TO TMP0
003144  011037  001162                          MOV     (R0),$REG0              ;MOVE FOR TYPEOUT REASONS
```

```
003150  104401  003156                  TYPE    ,71$            ;;TYPE ASCIZ STRING
003154  000403                          BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  / DEV/
003164                          70$:
003164  013746  001176                  MOV     $TMP0,-(SP)     ;;SAVE $TMP0 FOR TYPEOUT
003170  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
003172     002                          .BYTE   2               ;;TYPE 2 DIGIT(S)
003173     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
003174  104401  003202                  TYPE    ,73$            ;;TYPE ASCIZ STRING
003200  000402                          BR      72$             ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  / = /
003206                          72$:
003206  013746  001162                  MOV     $REG0,-(SP)     ;;SAVE $REG0 FOR TYPEOUT
003212  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
003214     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
003215     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
003216  062700  000014                  ADD     #14,R0          ;ADD 14 FOR NEXT ADDR
003222  023737  001176  001702          CMP     $TMP0,DEVCNT    ;SEE IF TMP0 = # OF DEVICES
003230  001343                          BNE     4$              ;IF NOT, GO TYPE NEXT ADDR
003232  104401  001223                  TYPE    ,$CRLF          ;TYPE <CR><LF>
003236  022737  000004  001702          CMP     #4,DEVCNT       ;SEE IF THERE ARE 4 DEVICES
003244  001002                          BNE     5$              ;IF NOT,SKIP THE TYPE OUT
003246  104401  011037                  TYPE    ,FOR4           ;ELSE TYPE MSSG FOR 4TH DEV
003252                          5$:



                                ;;*****************************************************************
                                ;*TEST 1         NO BUS GRANTS ISSUED WITH PROCESSOR AT HIGHER PRIORITY THAN BUS REQUEST
                                        ;*THIS TEST IS TO INSURE THAT ANY REQUEST IS NOT
                                        ;*HONORED AS LONG AS THE PROCESSOR IS AT THE SAME OR
                                        ;*HIGHER PRIORITY
                                ;;*****************************************************************
003252  000004                  TST1:   SCOPE
003254  004737  002566                  JSR     PC,CLRREG       ;CLEAR CONTENTS OF ALL AVAILABLE DEVS
003260                          NG:

003260  012777  004506  176374          MOV     #ERRCHK,@BE1VEC ;SET UP DEVICE 1 INTR VECTOR
003266  012777  000340  176370          MOV     #PR7,@BE1PSW    ;SET UP DEVICE 1 PSW VECTOR
003274  012737  002550  000004          MOV     #TYMOUT,ERRVEC  ;SET UP TRAP THRU LOC 4(TIME OUT VEC)
003302  012700  000340                  MOV     #PR7,R0         ;MOVE PS=7 TO R0
003306  012701  002021                  MOV     #2021,R1        ;MOVE FUN 1-DATI-BR7 TO R1
003312  004737  010416                  JSR     PC,NOG          ;DO  NOG
003316  012700  000300                  MOV     #PR6,R0         ;MOVE PS=6 TO R0
003322  012701  002011                  MOV     #2011,R1        ;MOVE FUN 1-DATI-BR6 TO R1
003326  004737  010416                  JSR     PC,NOG          ;DO NOG
003332  012700  000240                  MOV     #PR5,R0         ;MOVE PS=5 TO R0
003336  012701  002005                  MOV     #2005,R1        ;MOVE FUN 1-DATI-BR5 TO R1
003342  004737  010416                  JSR     PC,NOG          ;DO NOG
003346  012700  000200                  MOV     #PR4,R0         ;MOVE PS=4 TO R0
003352  012701  002003                  MOV     #2003,R1        ;MOVE FUN 1-DATI-BR4 TO R1
003356  004737  010416                  JSR     PC,NOG          ;DO NOG
003362  052777  004000  176220          BIS     #BIT11,@BE1CR1  ;SET BIT 11 TO DO FUN 3
003370  052777  000040  176212          BIS     #BIT05,@BE1CR1  ;SET OFF DEV AT NPR LEVEL
003376  000240                          NOP                     ;ALLOW TIME FOR XFER


```

```
                                          ;;*************************************************************************
                                          ;*TEST 2          ISSUING OF NON-PROCESSOR GRANTS AND ARBITRATION TESTS
                                                  ;*THIS TEST WILL REQUEST ON NPR THRU BR4 LEVELS
                                                  ;*WITH THE PROCESSOR STATUS INITIALLY AT LEVEL 7 AND MAKE
                                                  ;*SURE THE DEVICE EXERCISES AN NPG TO DO A FUN 1-DATI,
                                                  ;*THEN THE REQUESTS WILL BE REPEATED WHILE SEQUENTIALLY
                                                  ;*LOWERING THE PROCESSOR STATUS FROM 7 TO 0 TO ALLOW
                                                  ;*ARBITRATION OF ALL REQUESTS AND THE ISSUING OF NPG
                                          ;;*************************************************************************
 003400  000004                           TST2:   SCOPE

 003402                                   NPRTST:
 003402  012700  000340                           MOV     #PR7,R0
 003406                                   2$:
 003406  123737  001115  001103                   CMPB    $ERMAX,$ERFLG     ;MAX ERRS FOR THIS TEST OCCURRED?
 003414  100452                                   BMI     TST3              ;;BR IF YES TO NEXT TEST
 003416  012737  000340  177776                   MOV     #PR7,PSW                ;INITIAL PS
 003424  012777  004506  176230                   MOV     #ERRCHK,@BE1VEC   ;SET UP VECTOR LOCATION
 003432  012777  000340  176224                   MOV     #PR7,@BE1PSW      ;SET UP DEVICE INTR PSW
 003440  012777  020510  176140                   MOV     #ATEND,@BE1BA     ;SET UP ADDR REG
 003446  012777  177777  176130                   MOV     #-1,@BE1CC        ;SET CYCLE COUNT = 1
 003454  012777  002077  176126                   MOV     #2077,@BE1CR1     ;LOAD #2077 FUNTIONS
 003462  010037  177776                           MOV     R0,PSW            ;LOWER PROC STATUS
 003466  000240                                   NOP                       ;ALLOW TIME FOR INTERUPT

 003470  022777  177777  176106                   CMP     #-1,@BE1CC        ;SEE IF DEVICE WENT OFF
 003476  001014                                   BNE     5$                ;IF IT DID,SKIP ERR TYPEOUT
 003500  017737  176104  001162                   MOV     @BE1CR1,$REG0     ;NEXT MOVES ARE FOR TYPEOUTS
 003506  017737  176072  001164                   MOV     @BE1CC,$REG1
 003514  012737  000340  001166                   MOV     #PR7,$REG2
 003522  010037  001170                           MOV     R0,$REG3
 003526  104003                                   ERROR   3                 ;TYPE ERROR MESSG
 003530                                   5$:
 003530  162700  000040                           SUB     #40,R0            ;LOWER PS BY 1 LEVEL
 003534  020027  000000                           CMP     R0,#PR0 ;SEE IF R0 IS LESS THAN 0
 003540  100322                                   BPL     2$                ;IF PLUS ,GO BACK AND DO ANOTHER CYCLE

                                          ;;*************************************************************************
                                          ;*TEST 3          ISSUING OF BUS GRANT 7 AND ARBITRATION TESTS
                                                  ;*THIS TEST WILL ARBITRATE FOR A BG7,
                                                  ;*THE REQUESTS WILL BE ON LEVELS BR7 THRU BR4, DOING
                                                  ;*FUN 1-DATI TRANSFERS, AND THE PROCESSOR STATUS
                                                  ;*LOWERED SEQUENTIALLY FROM 7 TO 0.
                                          ;;*************************************************************************
 003542  000004                           TST3:   SCOPE
 003544                                   BR7TST:
 003544  012700  000300                           MOV     #PR6,R0           ;2ND PS WILL = 6
 003550                                   2$:
 003550  123737  001115  001103                   CMPB    $ERMAX,$ERFLG     ;MAX ERRS FOR THIS TEST OCCURRED?
 003556  100452                                   BMI     TST4              ;;BR IF YES TO NEXT TEST
 003560  012737  000340  177776                   MOV     #PR7,PSW                ;INITIAL PS
 003566  012777  004506  176066                   MOV     #ERRCHK,@BE1VEC   ;SET UP VECTOR LOCATION
 003574  012777  000340  176062                   MOV     #PR7,@BE1PSW      ;SET UP DEVICE INTR PSW
 003602  012777  020510  175776                   MOV     #ATEND,@BE1BA     ;SET UP ADDR REG
 003610  012777  177777  175766                   MOV     #-1,@BE1CC        ;SET CYCLE COUNT = 1
 003616  012777  002037  175764                   MOV     #2037,@BE1CR1     ;LOAD #2037 FUNTIONS
 003624  010037  177776                   MOV     R0,PSW              ;LOWER PROC STATUS
 003630  000240                           NOP                         ;ALLOW TIME FOR INTERUPT

 003632  022777  177777  175744           CMP     #-1,@BE1CC          ;SEE IF DEVICE WENT OFF
 003640  001014                           BNE     5$                  ;IF IT DID,SKIP ERR TYPEOUT
 003642  017737  175742  001162           MOV     @BE1CR1,$REG0       ;NEXT MOVES ARE FOR TYPEOUTS
 003650  017737  175730  001164           MOV     @BE1CC,$REG1
 003656  012737  000340  001166           MOV     #PR7,$REG2
 003664  010037  001170                   MOV     R0,$REG3
 003670  104004                           ERROR   4                   ;TYPE ERROR MESSG
 003672                          5$:
 003672  162700  000040                   SUB     #40,R0              ;LOWER PS BY 1 LEVEL
 003676  020027  000000                   CMP     R0,#PR0 ;SEE IF R0 IS LESS THAN 0
 003702  100322                           BPL     2$                  ;IF PLUS ,GO BACK AND DO ANOTHER CYCLE



                                 ;;*****************************************************************
                                 ;*TEST 4          ISSUING OF BUS GRANT 6 AND ARBITRATION TESTS
                                          ;*THIS TEST WILL ARBITRATE FOR A BG6,
                                          ;*THE REQUESTS WILL BE ON LEVELS BR6 THRU BR4, DOING
                                          ;*FUN 1-DATI TRANSFERS, AND THE PROCESSOR STATUS
                                          ;*LOWERED SEQUENTIALLY FROM 6 TO 0.
                                 ;;*****************************************************************
 003704  000004                  TST4:    SCOPE
 003706                          BR6TST:
 003706  012700  000240                   MOV     #PR5,R0             ;2ND PS WILL = 5
 003712                          2$:
 003712  123737  001115  001103           CMPB    $ERMAX,$ERFLG       ;MAX ERRS FOR THIS TEST OCCURRED?
 003720  100452                           BMI     TST5                ;;BR IF YES TO NEXT TEST
 003722  012737  000300  177776           MOV     #PR6,PSW                    ;INITIAL PS
 003730  012777  004506  175724           MOV     #ERRCHK,@BE1VEC ;SET UP VECTOR LOCATION
 003736  012777  000340  175720           MOV     #PR7,@BE1PSW        ;SET UP DEVICE INTR PSW
 003744  012777  020510  175634           MOV     #ATEND,@BE1BA       ;SET UP ADDR REG
 003752  012777  177777  175624           MOV     #-1,@BE1CC          ;SET CYCLE COUNT = 1
 003760  012777  002017  175622           MOV     #2017,@BE1CR1       ;LOAD #2017 FUNTIONS
 003766  010037  177776                   MOV     R0,PSW              ;LOWER PROC STATUS
 003772  000240                           NOP                         ;ALLOW TIME FOR INTERUPT

 003774  022777  177777  175602           CMP     #-1,@BE1CC          ;SEE IF DEVICE WENT OFF
 004002  001014                           BNE     5$                  ;IF IT DID,SKIP ERR TYPEOUT
 004004  017737  175600  001162           MOV     @BE1CR1,$REG0       ;NEXT MOVES ARE FOR TYPEOUTS
 004012  017737  175566  001164           MOV     @BE1CC,$REG1
 004020  012737  000300  001166           MOV     #PR6,$REG2
 004026  010037  001170                   MOV     R0,$REG3
 004032  104005                           ERROR   5                   ;TYPE ERROR MESSG
 004034                          5$:
 004034  162700  000040                   SUB     #40,R0              ;LOWER PS BY 1 LEVEL
 004040  020027  000000                   CMP     R0,#PR0 ;SEE IF R0 IS LESS THAN 0
 004044  100322                           BPL     2$                  ;IF PLUS ,GO BACK AND DO ANOTHER CYCLE



                                 ;;*****************************************************************
                                 ;*TEST 5          ISSUING OF BUS GRANT 5 AND ARBITRATION TESTS
                                          ;*THIS TEST WILL ARBITRATE FOR A BG5,
                                          ;*THE REQUESTS WILL BE ON LEVELS BR5 THRU BR4, DOING
                                          ;*FUN 1-DATI TRANSFERS, AND THE PROCESSOR STATUS
```

```
                                                 ;*LOWERED SEQUENTIALLY FROM 5 TO 0.
                                        ;;*****************************************************************
 004046  000004                         TST5:    SCOPE
 004050                                 BR5TST:
 004050  012700  000200                          MOV     #PR4,R0          ;2ND PS WILL = 4
 004054                                 2$:
 004054  123737  001115  001103                  CMPB    $ERMAX,$ERFLG    ;MAX ERRS FOR THIS TEST OCCURRED?
 004062  100452                                  BMI     TST6             ;;BR IF YES TO NEXT TEST
 004064  012737  000240  177776                  MOV     #PR5,PSW                 ;INITIAL PS
 004072  012777  004506  175562                  MOV     #ERRCHK,@BE1VEC  ;SET UP VECTOR LOCATION
 004100  012777  000340  175556                  MOV     #PR7,@BE1PSW     ;SET UP DEVICE INTR PSW
 004106  012777  020510  175472                  MOV     #ATEND,@BE1BA    ;SET UP ADDR REG
 004114  012777  177777  175462                  MOV     #-1,@BE1CC       ;SET CYCLE COUNT = 1
 004122  012777  002007  175460                  MOV     #2007,@BE1CR1    ;LOAD #2007 FUNTIONS
 004130  010037  177776                          MOV     R0,PSW           ;LOWER PROC STATUS
 004134  000240                                  NOP                      ;ALLOW TIME FOR INTERUPT

 004136  022777  177777  175440                  CMP     #-1,@BE1CC       ;SEE IF DEVICE WENT OFF
 004144  001014                                  BNE     5$               ;IF IT DID,SKIP ERR TYPEOUT
 004146  017737  175436  001162                  MOV     @BE1CR1,$REG0    ;NEXT MOVES ARE FOR TYPEOUTS
 004154  017737  175424  001164                  MOV     @BE1CC,$REG1
 004162  012737  000240  001166                  MOV     #PR5,$REG2
 004170  010037  001170                          MOV     R0,$REG3
 004174  104006                                  ERROR   6                ;TYPE ERROR MESSG
 004176                                 5$:
 004176  162700  000040                          SUB     #40,R0           ;LOWER PS BY 1 LEVEL
 004202  020027  000000                          CMP     R0,#PR0 ;SEE IF R0 IS LESS THAN 0
 004206  100322                                  BPL     2$               ;IF PLUS ,GO BACK AND DO ANOTHER CYCLE


                                        ;;*****************************************************************
                                        ;*TEST 6          ISSUING OF BUS GRANT 4 AND ARBITRATION TESTS
                                                 ;*THIS TEST WILL ARBITRATE FOR A BG4,
                                                 ;*THE REQUESTS WILL BE ON LEVEL BR4, DOING
                                                 ;*FUNC 1-DATI TRANSFERS, AND THE PROCESSOR STATUS
                                                 ;*LOWERED SEQUENTIALLY FROM 4 TO 0.
                                        ;;*****************************************************************
 004210  000004                         TST6:    SCOPE
 004212                                 BR4TST:
 004212  012700  000140                          MOV     #PR3,R0          ;2ND PS WILL = 3
 004216                                 2$:
 004216  123737  001115  001103                  CMPB    $ERMAX,$ERFLG    ;MAX ERRS FOR THIS TEST OCCURRED?
 004224  100452                                  BMI     TST7             ;;BR IF YES TO NEXT TEST
 004226  012737  000200  177776                  MOV     #PR4,PSW                 ;INITIAL PS
 004234  012777  004506  175420                  MOV     #ERRCHK,@BE1VEC  ;SET UP VECTOR LOCATION
 004242  012777  000340  175414                  MOV     #PR7,@BE1PSW     ;SET UP DEVICE INTR PSW
 004250  012777  020510  175330                  MOV     #ATEND,@BE1BA    ;SET UP ADDR REG
 004256  012777  177777  175320                  MOV     #-1,@BE1CC       ;SET CYCLE COUNT = 1
 004264  012777  002003  175316                  MOV     #2003,@BE1CR1    ;LOAD #2003 FUNTIONS
 004272  010037  177776                          MOV     R0,PSW           ;LOWER PROC STATUS
 004276  000240                                  NOP                      ;ALLOW TIME FOR INTERUPT

 004300  022777  177777  175276                  CMP     #-1,@BE1CC       ;SEE IF DEVICE WENT OFF
 004306  001014                                  BNE     5$               ;IF IT DID,SKIP ERR TYPEOUT
 004310  017737  175274  001162                  MOV     @BE1CR1,$REG0    ;NEXT MOVES ARE FOR TYPEOUTS
 004316  017737  175262  001164                  MOV     @BE1CC,$REG1
```

```
004324  012737  000200  001166          MOV     #PR4,$REG2
004332  010037  001170                  MOV     R0,$REG3
004336  104007                          ERROR   7                       ;TYPE ERROR MESSG
004340                          5$:
004340  162700  000040                  SUB     #40,R0                  ;LOWER PS BY 1 LEVEL
004344  020027  000000                  CMP     R0,#PR0 ;SEE IF R0 IS LESS THAN 0
004350  100322                          BPL     2$                      ;IF PLUS ,GO BACK AND DO ANOTHER CYCLE
                                ;;*****************************************************************


                                ;;*****************************************************************
                                ;*TEST 7         CPU TEST FOR NO SACK TIME OUT
                                        ;*THIS TEST WILL CHECK THAT THE CPU TIMES OUT AND
                                        ;*DROPS A GRANT IF NO SACK SIGNAL IS RECEIVED
                                        ;*IF THE CPU TIME OUT IS INOPERATIVE, THE BUS EXERCISER
                                        ;*WILL TIME OUT AND SEND THE SACK SIGNAL TO PREVENT
                                        ;*A BUS HANG AND SET AN ERROR FLAG IN CR2
                                ;;*****************************************************************
004352  000004                  TST7:   SCOPE
004354  004737  002566                  JSR     PC,CLRREG               ;CLEAR CONTENTS OF ALL AVAILABLE DEVS
004360  012777  177777  175216          MOV     #-1,@BE1CC              ;SET CYCLE COUNT = 1
004366  012777  020510  175212          MOV     #ATEND,@BE1BA           ;SET UP DEVICE REG ADDR
004374  012737  000340  177776          MOV     #PR7,PSW                ;SET PS=7
004402  012737  002550  000004          MOV     #TYMOUT,ERRVEC          ;SET UP TIME OUT VECTOR
004410  012777  004506  175244          MOV     #ERRCHK,@BE1VEC         ;SET UP DEVICE INTR VECTOR
004416  012777  000340  175240          MOV     #PR7,@BE1PSW            ;SET UP DEVICE INTR PSW
004424  052777  000010  175162          BIS     #BIT03,@BE1CR2          ;INHIBIT SACK RETURN
004432  012777  006003  175150          MOV     #6003,@BE1CR1           ;DO FUN 3--BR4
004440  012737  000140  177776          MOV     #PR3,PSW                ;LOWER PROC. STATUS TO 3
004446  004737  010626                  JSR     PC,CNTR                 ;DELAY FOR TIMEOUT
004452  042777  000010  175134          BIC     #BIT03,@BE1CR2          ;ALLOW FUTURE SACKS
004460  105777  175130                  TSTB    @BE1CR2                 ;CHECK IF NO-NO SACK BIT IS SET
004464  100024                          BPL     TST10                   ;;IF NOT SET, GO TO NEXT TEST
004466  017737  175116  001162          MOV     @BE1CR1,$REG0           ;MOVE FOR TYPEOUT REASONS
004474  017737  175114  001164          MOV     @BE1CR2,$REG1           ;MOVE FOR TYPEOUT
004502  104012                          ERROR   12                      ;ERROR IF SET-DEVICE FORCED TO SEND SACK
004504  000414                          BR      TST10                   ;;GO TO NEXT TEST
                                ;;*****************************************************************
                                ;;*****************************************************************
004506                          ERRCHK:
004506  033777  001576  175100          BIT     ALLERR,@BE1CR2 ;CHECK FOR ANY ERRS IN CR2
004514  001407                          BEQ     5$                      ;IF NONE, EXIT
004516  011637  001174                  MOV     (SP),$REG5              ;FOR TYPEOUT OF PC
004522  012737  000001  001206          MOV     #1,$TMP4                ;INDICATOR FOR DEVICE 1
004530  004737  010166                  JSR     PC,ERRTN                ;CHECK TO SEE IF ANY ERRORS OCCURED
004534                          5$:
004534  000002                          RTI                             ;EXIT TRAP


                                ;;*****************************************************************
                                ;*TEST 10        CPU TEST FOR RECEIVING SACK
                                        ;*THIS TEST IS TO INSURE THAT THE CPU CAN RECEIVE THE
                                        ;*SACK SIGNAL; THE TIME DELAY WILL BE SET ON DEVICE 1
                                        ;*AND SEVERAL DATI TRANSFERS MADE, IF THERE IS NO BUS
                                        ;*LATE ERROR, THE CPU RECEIVED SACK CORRECTLY
                                        ;*IT IS ASSUMED THAT DEV 1 TIME DELAY IS SET FOR 10 US
                                ;;*****************************************************************
```

```
004536  000004                          TST10:  SCOPE

004540  012737  000340  177776                  MOV     #PR7,PSW         ;PS = 7
004546  004737  002566                          JSR     PC,CLRREG        ;CLEAR ALL DEVICE REGISTERS
004552  012702  020510                          MOV     #ATEND,R2        ;R2 WILL POINT TO END OF PROG
004556  012705  000010                          MOV     #10,R5           ;R5 = # OF TEST WORDS TO CREATE
004562  004737  010610                          JSR     PC,DOUP          ;CREATE THOSE TEST WORDS

004566  012777  004506  175066                  MOV     #ERRCHK,@BE1VEC  ;SET UP VECTOR LOCATION
004574  012777  000340  175062                  MOV     #PR7,@BE1PSW     ;SET UP DEVICE INTR PSW
004602  012777  177770  174774                  MOV     #-10,@BE1CC      ;SET UP CYCLE COUNT
004610  012777  020510  174770                  MOV     #ATEND,@BE1BA    ;SET UP ADDR REGISTER
004616  052777  040000  174770                  BIS     #BIT14,@BE1CR2   ;SET BIT 14 OF CR2 FOR TIME DELAY
004624  012777  024441  174756                  MOV     #24441,@BE1CR1   ;DO FUN 2-DATIP/NO ROL-NPR
004632  012737  000000  177776                  MOV     #PR0,PSW         ;LOWER PS TO ALLOW INTERRUPTS
004640                                  5$:
004640  000240                                  NOP                      ;ALLOW FOR INTERRUPT
004642  105777  174742                          TSTB    @BE1CR1          ;SEE IF DONE BIT SET
004646  100374                                  BPL     5$               ;IF NOT, GO BACK AND WAIT
004650  042777  040000  174736                  BIC     #BIT14,@BE1CR2   ;ELSE CLEAR BIT 14 OF CR2
004656  022777  000010  174716                  CMP     #10,@BE1DB       ;DID LAST XFER MOVE 10 INTO DB
004664  001407                                  BEQ     10$              ;IF IT DID,GO TO 10$
004666  017737  174710  001164                  MOV     @BE1DB,$REG1     ;ELSE MOVE FOR ERR TYPE OUT
004674  012737  000010  001166                  MOV     #10,$REG2
004702  104034                                  ERROR   34               ;TYPE ERR MSSG
004704                                  10$:
004704  032777  004000  174702                  BIT     #BIT11,@BE1CR2   ;SEE IF NO SSYN ON INTR ERR SET
004712  001402                                  BEQ     TST11            ;;IF NOT SET, GO TO NEXT TEST
004714  104023                                  ERROR   23               ;ELSE TYPE ERR MSSG
004716  000400                                  BR      TST11            ;;THEN GO TO NEXT TEST



                                        ;;*****************************************************************
                                        ;*TEST 11        PASSING OF GRANTS AND INTERRUPT TEST
                                                ;*THIS TEST WILL SET OFF ALL AVAILABLE DEVICES SIMULTANEOUSLY
                                                ;*WHOSE ONLY FUNCTIONS WILL BE TO INTERRUPT, THE REQUESTS
                                                ;*WILL ALL BE AT LEVEL 7 SO THAT THE DEVICE CLOSEST TO THE CPU
                                                ;*SHOULD RECEIVE BG7 FIRST AND INTERRUPT FIRST, THE NEXT
                                                ;*CLOSEST SHOULD INTERRUPT NEXT AND SO ON.
                                        ;;*****************************************************************
004720  000004                          TST11:  SCOPE
004722  012737  000340  177776                  MOV     #PR7,PSW         ;PS=7
004730  004737  002566                          JSR     PC,CLRREG        ;CLEAR CONTENTS OF ALL AVAILABLE DEVS
004734                                  LOAD1:
004734  012704  001704                          MOV     #DEVS,R4         ;DEVS CONTAINS SEQUENCE OF INTR'G DEVICE ADDRS
004740  012777  005162  174714                  MOV     #INTR1,@BE1VEC   ;SET UP DEVICE 1 INTR VECTOR
004746  012777  000340  174710                  MOV     #PR7,@BE1PSW     ;SET UP INTR PSW
004754  012777  000036  174626                  MOV     #36,@BE1CR1      ;DO FUN 0 - BR7 THRU BR4
004762  122737  000001  001702                  CMPB    #1,DEVCNT        ;IF ONLY 1 DEVICE ON BUS
004770  001443                                  BEQ     GO               ;BRANCH TO GO
004772  012777  005200  174666          LOAD2:  MOV     #INTR2,@BE2VEC   ;SET UP DEVICE 2 INTR VECTOR
005000  012777  000340  174662                  MOV     #PR7,@BE2PSW     ;SET UP DEVICE 2 PSW VECTOR
005006  012777  000036  174610                  MOV     #36,@BE2CR1      ;DO FUN 0 - BR7 THRU BR4
005014  122737  000002  001702                  CMPB    #2,DEVCNT        ;IF ONLY 2 DEVICES ON BUS
005022  001426                                  BEQ     GO               ;BRANCH TO GO
005024  012777  005216  174640          LOAD3:  MOV     #INTR3,@BE3VEC   ;SET UP DEVICE 3 INTR VECTOR
```

```
005032  012777  000340  174634           MOV     #PR7,@BE3PSW     ;SET UP DEVICE 3 PSW VECTOR
005040  012777  000036  174572           MOV     #36,@BE3CR1      ;DO FUN 0 - BR7 THRU BR4
005046  122737  000003  001702           CMPB    #3,DEVCNT        ;IF ONLY 3 DEVICES ON BUS
005054  001411                           BEQ     GO               ;BRANCH TO GO
005056                          LOAD4:
005056  012777  005234  174612           MOV     #INTR4,@BE4VEC   ;SET UP DEVICE 4 INTR VECTOR
005064  012777  000340  174606           MOV     #PR7,@BE4PSW     ;SET UP DEVICE 4 PSW VECTOR
005072  012777  000036  174554           MOV     #36,@BE4CR1      ;DO FUN 0 - BR7 THRU BR4
005100                          GO:
005100  005001                           CLR     R1               ;CLEAR R1 FOR COUNTING
005102  005277  174472                   INC     @SIMLGO          ;SET SIMULTANEOUS GO REGISTER
005106  012737  000000  177776           MOV     #PR0,PSW         ;LOWER PS TO ALLOW INTERRUPTS
005114  004737  010626                   JSR     PC,CNTR          ;ALLOW TIME FOR INTERRUPTS BY COUNTING
005120  020137  001702          CMPARE:  CMP     R1,DEVCNT        ;COMPARE THE TWO
005124  001456                           BEQ     TST12            ;;IF BUFFERS INCREMENTED IN CORRECT SEQUENCE, GO TO NEXT
005126  013737  001704  001164           MOV     DEVS,$REG1       ;MOVE FOR TYPEOUT REASONS
005134  013737  001706  001166           MOV     DEVS+2,$REG2     ;MOVE FOR TYPEOUT REASONS
005142  013737  001710  001170           MOV     DEVS+4,$REG3     ;MOVE FOR TYPEOUT REASONS
005150  013737  001712  001172           MOV     DEVS+6,$REG4     ;MOVE FOR TYPEOUT REASONS
005156  104010                           ERROR   10               ;TYPE ERR MSSG
005160  000440                           BR      TST12            ;;GO TO NEXT TEST
                                ;;*****************************************************************
                                ;;*****************************************************************
005162                          INTR1:
005162  005201                           INC     R1               ;ADD 1 TO COUNTER ON INTR
005164  013724  001602                   MOV     BE1DB,(R4)+      ;MOVE ADDR FOR TYPEOUT
005170  012737  000001  001206           MOV     #1,$TMP4         ;INDICATOR FOR DEVICE 1
005176  000424                           BR      INTRTN           ;BRANCH TO REST OF INTR RTN
005200                          INTR2:
005200  005201                           INC     R1               ;ADD 1 TO COUNTER ON INTR
005202  013724  001616                   MOV     BE2DB,(R4)+      ;MOVE ADDR FOR TYPEOUT
005206  012737  000002  001206           MOV     #2,$TMP4         ;INDICATOR FOR DEVICE 2
005214  000415                           BR      INTRTN           ;BRANCH TO REST OF INTR RTN
005216                          INTR3:
005216  005201                           INC     R1               ;ADD 1 TO COUNTER ON INTR
005220  013724  001632                   MOV     BE3DB,(R4)+      ;MOVE ADDR FOR TYPEOUT
005224  012737  000003  001206           MOV     #3,$TMP4         ;INDICATOR FOR DEVICE 3
005232  000406                           BR      INTRTN           ;BRANCH TO REST OF INTR RTN
005234                          INTR4:
005234  005201                           INC     R1               ;ADD 1 TO COUNTER ON INTR
005236  013724  001646                   MOV     BE4DB,(R4)+      ;MOVE ADDR FOR TYPEOUT
005242  012737  000004  001206           MOV     #4,$TMP4         ;INDICATOR FOR DEVICE 4
005250                          INTRTN:
005250  011637  001174                   MOV     (SP),$REG5       ;FOR TYPEOUT OF PC
005254  004737  010166                   JSR     PC,ERRTN         ;SEE IF ERROR CAUSED INTR
005260  000002                           RTI                      ;EXIT



                                ;;*****************************************************************
                                ;*TEST 12        ADDRESS LINES (14 - 17) CHECK
                                         ;*THIS TEST WILL CHECK BUS ADDRESS LINES 14 THRU 17
                                         ;*BY DOING A FUN 1-DATI-NPR TO THOSE ADDRESSES
                                         ;*IF THE ADDRESSES DON'T EXIST THE INTERRUPT ROUTINE
                                         ;*WILL IGNORE ANY NO SSYN ERROR.
                                ;;*****************************************************************
005262  000004                  TST12:   SCOPE
```

```

005264  004737  002566                  JSR     PC,CLRREG          ;CLEAR CONTENTS OF ALL AVAILABLE DEVS
005270  012737  000140  177776          MOV     #PR3,PSW           ;PS=3
005276  012777  005364  174356          MOV     #BRK,@BE1VEC       ;SET UP DEVICE INTR VEC
005304  012777  000340  174352          MOV     #PR7,@BE1PSW       ;SET UP DEVICE PSW VEC
005312                          DO14:
005312  004737  010654                  JSR     PC,ADLI            ;TEST ADDR LINES 14 &15
005316  052777  000001  174270          BIS     #1,@BE1CR2         ;ELSE SET BIT 0 OF CR2(ADDR LINE 16)
005324  004737  010654                  JSR     PC,ADLI
005330  042777  000001  174256          BIC     #1,@BE1CR2         ;CLEAR BIT 0(ADDR LINE 16)
005336  052777  000002  174250          BIS     #2,@BE1CR2         ;SET BIT 1 OF CR2(ADDR LINE 17)
005344  004737  010654                  JSR     PC,ADLI
005350  052777  000003  174236          BIS     #3,@BE1CR2         ;ELSE SET BITS 0 AND 1 OF CR2
                                                                   ;SETS ADDR LINES 16 & 17
005356  004737  010654                  JSR     PC,ADLI
005362  000431                          BR      TST13              ;;GO TO NEXT TEST
                                ;;*************************************************************
                                ;;*************************************************************
005364                          BRK:
005364  011637  001174                  MOV     (SP),$REG5         ;FOR TYPEOUT OF PC
005370  012737  000001  001206          MOV     #1,$TMP4           ;INDICATOR FOR DEVICE 1
005376  032777  007340  174210          BIT     #7340,@BE1CR2      ;CHECK FOR ALL ERRS EXCEPT NO SSYN ERR
005404  001003                          BNE     1$                 ;IF ANY ARE SET,SEE WHICH ONES
005406  005077  174200                  CLR     @BE1CLR            ;ELSE CLEAR THE NO SSYN ERR
005412  000414                          BR      EXBRK              ;AND EXIT
005414                          1$:
005414  017737  174170  001164          MOV     @BE1CR1,$REG1      ;MOVES ARE FOR TYPEOUTS
005422  017737  174166  001166          MOV     @BE1CR2,$REG2
005430  017737  174152  001170          MOV     @BE1BA,$REG3
005436  104011                          ERROR   11                 ;ERR ON ACCESSING A14 - A17
005440  004737  010166                  JSR     PC,ERRTN           ;DO ERR CHECK SUB-ROUTINE
005444  000002                  EXBRK:  RTI                        ;EXIT


                                ;;*************************************************************
                                ;*TEST 13       CPU TEST FOR ACLO/DCLO SEQUENCE
                                        ;*THIS TEST CHECKS THE ASSERTION OF ACLO AND DCLO
                                        ;*AND THAT THE CPU TRAPS TO THE CORRECT SERVICE ROUTINE.
                                        ;*IF THIS PROGRAM IS RUNNING UNDER ACT11 THIS TEST
                                        ;*WILL BE SKIPPED.
                                ;;*************************************************************
005446  000004                  TST13:  SCOPE
005450  012737  000001  001212          MOV     #1,$TIMES          ;;DO 1 ITERATION
005456  005737  000042                  TST     42                 ;SEE IF PROGRAM IS UNDER ACT11
005462  001061                          BNE     TST14              ;;IF UNDER ACT, DO NOT PERFORM THIS TEST
005464  012705  000001                  MOV     #1,R5              ;INIT R5 WITH A VALUE OF 1
005470                          6$:
005470  005205                          INC     R5                 ;ADD 1 TO R5
005472  100376                          BPL     6$                 ;KEEP ADDING AS LONG AS R5 POS
005474  012737  000001  001206          MOV     #1,$TMP4           ;INDICATOR FOR DEVICE 1
005502  012737  000340  177776          MOV     #PR7,PSW           ;SET PS=7
005510  012777  004506  174144          MOV     #ERRCHK,@BE1VEC    ;SET UP INTR VECTOR
005516  012777  000340  174140          MOV     #PR7,@BE1PSW       ;SET UP DEVICE INTR PSW
005524  005037  001200                  CLR     $TMP1              ;CLEAR TEMPORARY REGISTER(TMP1)
005530  012737  005606  000024          MOV     #TMPPWR,PWRVEC     ;SET UP SPECIAL POWER RTN
005536  052777  000020  174050          BIS     #BIT04,@BE1CR2     ;INDICATE PWR FAILURE BY SETTING BIT 4
```

```
005544  004737  010626                  JSR     PC,CNTR         ;PAUSE FOR TIME
005550  012737  010122  000024          MOV     #PWRFAL,PWRVEC  ;RESTORE PWRFAL SEQ FOR A PWR FAIL
005556  042777  000020  174030          BIC     #BIT04,@BE1CR2  ;MAKE SURE BIT 4 IS CLEARED
005564                          FAILCK:
005564  022737  001154  020466          CMP     #$NULL,$PWRMG   ;IF THIS TEST IS CAUSE OF
                                                                ;PWR FAIL --TYPE NULL CHAR
005572  001401                          BEQ     XTST            ;IF EQUAL, EXIT TEST
005574  104013                          ERROR   13              ;TYPE ERR MSSG IF FAILURE
005576                          XTST:
005576  012737  020500  020466          MOV     #$POWER,$PWRMG  ;RESTORE TYPE OUT OF 'POWER'
005604  000410                          BR      TST14           ;;GO TO NEXT TEST
                                ;;*************************************************************
                                ;;*************************************************************
005606                          TMPPWR:                         ;SPECIAL PWR RTN; OTHER THAN SYSMAC'S
005606  012737  001154  020466          MOV     #$NULL,$PWRMG   ;CHANGE PWR MSSG TO NULL CHAR
005614  042777  000020  173772          BIC     #BIT04,@BE1CR2  ;CLEAR POWER DOWN/UP BIT
005622  000137  020332                  JMP     $PWRDN          ;GO TO THAT RTN



                                ;;*************************************************************
                                ;*TEST 14        PARITY ERROR TEST
                                        ;*THIS TEST WILL CAUSE PARITY ERROR AND CHECKS
                                        ;*THAT THE CPU TRAPS TO THE CORRECT VECTOR.
                                        ;*THIS TEST SHOULD BE DESELECTED IF THE MEMORY
                                        ;*PARITY OPTION IS NOT PRESENT, ELSE AN
                                        ;*ERROR WILL BE REPORTED ALTHOUGH HARDWARE IS
                                        ;*FUNCTIONING PROPERLY.
                                        ;*SW06=1         INHIBIT TEST 14 AND GO TO NEXT TEST
                                ;;*************************************************************
005626  000004                  TST14:  SCOPE
005630  032777  000100  173302          BIT     #BIT06, @SWR    ;INHIBIT TEST 14?
005636  001105                          BNE     TST15           ;GO TO NEXT TEST
005640  012737  006050  000010          MOV     #NODO,10        ;SET UP RESERVED INSTR VECTOR
005646  012737  000340  000012          MOV     #PR7,12         ;PSW=7
005654  005037  001204                  CLR     $TMP3           ;SET $TMP3 = 0
005660  000270                          SEN                     ;SET N BIT OF CC
005662  006737  001204                  SXT     $TMP3           ;IF VALID INSTR, $TMP3 WILL = -1
005666  005737  001204                  TST     $TMP3           ;IF INVALID, $TMP3 WILL REMAIN 0
005672  100033                          BPL     NXT             ;IF CP NOT= 35,40,45,OR 70,GO TO NEXT TEST
005674  012737  000140  177776          MOV     #PR3,PSW        ;PS=3
005702  012777  006014  173752          MOV     #PBERR,@BE1VEC  ;SET UP DEVICE INTR
005710  012737  006036  000114          MOV     #PBRTN,PBVEC    ;SET UP PARITY BIT VECTOR
005716  012737  000340  000116          MOV     #340,PBPSW      ;SET UP PARITY BIT PSW
005724  012777  020510  173654          MOV     #ATEND,@BE1BA   ;SET UP ADDR REG
005732  012777  177777  173644          MOV     #-1,@BE1CC      ;SET UP CYCLE COUNT
005740  052777  010000  173646          BIS     #BIT12,@BE1CR2  ;SET BIT 12 FOR PARITY ERROR
005746  005777  173642                  TST     @BE1CR2         ;SET OFF PARITY ERR SEQUENCE
005752  012777  013161  173630          MOV     #13161,@BE1CR1  ;TRY FUN 1-DATO FROM CC-NPR-INTR ON DONE(7)
005760  000240                          NOP                     ;ALLOW TIME FOR ATTEMPTED XFER
005762                          NXT:
005762  012737  000116  000114          MOV     #PBPSW,PBVEC    ;RESTORE
005770  012737  000000  000116          MOV     #0,PBPSW        ;TRAP CATCHER HERE AND
005776  012737  000012  000010          MOV     #12,10          ;AT RESERVED
006004  012737  000000  000012          MOV     #0,12           ;INSTRUCTION VECTOR
006012  000417                          BR      TST15           ;;BRANCH TO NEXT TEST IF PARITY TRAP OCCURRED
006014                          PBERR:
```

```
006014  011637  001174                     MOV     (SP),$REG5       ;FOR TYPEOUT OF PC
006020  104014                             ERROR   14               ;TYPE ERR MSSG IF DEVICE INTERRUPTED
006022  012737  000001  001206             MOV     #1,$TMP4         ;INDICATOR FOR DEVICE 1
006030  004737  010166                     JSR     PC,ERRTN         ;CHECK TO SEE IF ANY ERRORS OCCURED
006034  000002                             RTI                      ;EXIT TRAP
                                   ;;***********************************************************
                                   ;;***********************************************************
006036                             PBRTN:                           ;PARITY BIT TRAP RTN
006036  012777  000000  173550             MOV     #0,@BE1CR2       ;CLEAR PARITY BIT ERROR-MUST BE DONE
                                                                    ;BY MOVING 0(S) TO BE1CR2
006044  012716  005762                     MOV     #NXT,(SP)        ;SET STACK FOR NEXT TEST
006050  000002                     NODO:   RTI



                                   ;;***********************************************************
                                   ;*TEST 15          MULTITRANSFERS I
                                           ;*THIS TEST WILL CAUSE ANY BUS EXERCISERS, UP TO 4,
                                           ;*TO CREATE A LOT OF TRAFFIC ON THE BUS AND
                                           ;*CHECK THAT THE CPU CAN HANDLE IT; ALL DEVICES
                                           ;*ARE SET OFF SIMULTANEOUSLY
                                   ;;***********************************************************
006052  000004                     TST15:  SCOPE
006054  004737  002566                     JSR     PC,CLRREG        ;CLEAR CONTENTS OF ALL AVAILABLE DEVS
006060  012703  000000                     MOV     #0,R3            ;SET DATA PATTERN = 0
006064  012704  177777                     MOV     #177777,R4       ;SET DATA PATTERN = ALL 1'S
006070  004737  007260                     JSR     PC,MULT1         ;LOAD & EXECUTE ALL DEVICES
006074  022737  000002  001702             CMP     #2,DEVCNT        ;ARE THERE MORE THAN 2 DEVICES?
006102  100115                             BPL     TST16            ;;IF 2 OR LESS, GO TO NEXT TEST
006104  012703  161610                     MOV     #161610,R3       ;ELSE LOAD R3 AND R4 WITH
006110  012704  016161                     MOV     #016161,R4       ;ANOTHER PATTERN
006114                             5$:
006114  123737  001115  001103             CMPB    $ERMAX,$ERFLG    ;MAX ERRS FOR THIS TEST OCCURRED?
006122  100505                             BMI     TST16            ;;BR IF YES TO NEXT TEST
006124  004737  010360                     JSR     PC,ROTATE        ;ROTATE DATA PATTERNS
006130  004737  007260                     JSR     PC,MULT1         ;LOAD & EXECUTE ALL DEVICES
006134  022703  107070                     CMP     #107070,R3       ;IS R3 = 107070?
006140  001365                             BNE     5$               ;IF NOT,ROTATE AND DO AGAIN
006142  012703  167777                     MOV     #167777,R3       ;ELSE MOVE NEW PATTERNS
006146  012704  010000                     MOV     #010000,R4       ;INTO R3 AND R4
006152                             10$:
006152  123737  001115  001103             CMPB    $ERMAX,$ERFLG    ;MAX ERRS FOR THIS TEST OCCURRED?
006160  100466                             BMI     TST16            ;;BR IF YES TO NEXT TEST
006162  004737  010360                     JSR     PC,ROTATE        ;ROTATE DATA PATTERNS
006166  004737  007260                     JSR     PC,MULT1         ;LOAD & EXECUTE ALL DEVICES
006172  022703  167777                     CMP     #167777,R3       ;IS R3 = 167777 AGAIN?
006176  001365                             BNE     10$              ;IF NOT,ROTATE AND DO AGAIN
006200  000456                             BR      TST16            ;;GO TO NEXT TEST
                                   ;;***********************************************************
                                   ;;***********************************************************
                                   ;;***********************************************************
006202                             SERV1:
006202  017737  173400  001714             MOV     @BE1BA,DATA1     ;MOVE ADDR IN BE1BA TO DATA1 AND
006210  162737  000001  001714             SUB     #1,DATA1         ;SUB 1 TO GET ACTUAL ADDR
006216  012737  000001  001206             MOV     #1,$TMP4         ;INDICATOR FOR DEVICE 1
006224  000435                             BR      INK              ;BRANCH TO INK
006226                             SERV2:
```

```
006226  017737  173370  001716          MOV     @BE2BA,DATA2        ;MOVE ADDR IN BE2BA TO DATA2 AND
006234  162737  000002  001716          SUB     #2,DATA2            ;SUB 2 TO GET ACTUAL ADDR
006242  012737  000002  001206          MOV     #2,$TMP4            ;INDICATOR FOR DEVICE 2
006250  000423                          BR      INK                 ;BRANCH TO INK
006252                          SERV3:
006252  017737  173360  001720          MOV     @BE3BA,DATA3        ;MOVE ADDR IN BE3BA TO DATA3 AND
006260  162737  000002  001720          SUB     #2,DATA3            ;SUB 2 TO GET ACTUAL ADDR
006266  012737  000003  001206          MOV     #3,$TMP4            ;INDICATOR FOR DEVICE 3
006274  000411                          BR      INK                 ;BRANCH TO INK
006276                          SERV4:
006276  017737  173350  001722          MOV     @BE4BA,DATA4        ;MOVE ADDR IN BE4BA TO DATA4 AND
006304  162737  000002  001722          SUB     #2,DATA4            ;SUB 2 TO GET ACTUAL ADDR
006312  012737  000004  001206          MOV     #4,$TMP4            ;INDICATOR FOR DEVICE 4
006320                          INK:
006320  005237  001166                  INC     $REG2               ;INCREMENT REG
006324  011637  001174                  MOV     (SP),$REG5          ;FOR TYPEOUT OF PC
006330  004737  010166                  JSR     PC,ERRTN            ;CHECK FOR ANY ERRS
006334  000002                          RTI                         ;EXIT
                                ;;*************************************************************************

                                ;;*************************************************************************
                                ;*TEST 16          MULTITRANSFERS II
                                        ;*THIS TEST WILL HAVE THE AVAILABLE EXERCISERS DOING
                                        ;*VARIOUS TRANSFERS AND/OR INTERRUPTS AT DIFFERENT
                                        ;*REQUEST LEVELS TO FURTHER CHECK CPU HANDLING CAPABILITIES
                                ;;*************************************************************************
006336  000004                  TST16:  SCOPE
006340  012702  020510                  MOV     #ATEND,R2           ;R2 = END OF PROG
006344  012705  005000                  MOV     #5000,R5            ;R5 = THE # OF DATA WORDS
006350  004737  010610                  JSR     PC,DOUP             ;CREATE THOSE WORDS IN BUFFER MEMORY
006354  004737  010722                  JSR     PC,TSTOVR           ;SET UP PATTERN IN MEMORY BUFFER AREA
006360  004737  002566                  JSR     PC,CLRREG           ;CLEAR CONTENTS OF ALL AVAILABLE DEVS

006364  012737  000000  177776          MOV     #PR0,PSW            ;PS=0
006372  012777  007124  173262          MOV     #S1,@BE1VEC         ;SET UP DEVICE 1 INTR VECTOR
006400  012777  000340  173256          MOV     #PR7,@BE1PSW        ;SET UP DEVICE 1 PSW VECTOR
006406  012777  022510  173172          MOV     #ATEND+2000,@BE1BA  ;SET UP ADDR REG
006414  012777  176000  173162          MOV     #-2000,@BE1CC       ;SET UP CYCLE COUNT
006422  012777  015551  173160          MOV     #15551,@BE1CR1      ;DO FUN 2-DATOB FROM CC-NPR-INTR ON DONE(6)
006430  022737  000001  001702          CMP     #1,DEVCNT           ;CHECK FOR MORE THAN 1 DEVICE
006436  001467                          BEQ     1$                  ;IF NOT, GO CHECK RESULTS

006440  012777  007156  173220          MOV     #S2,@BE2VEC         ;SET UP DEVICE 2 INTR VECTOR
006446  012777  000340  173214          MOV     #PR7,@BE2PSW        ;SET UP DEVICE 2 PSW VECTOR
006454  012777  177000  173136          MOV     #-1000,@BE2CC       ;SET UP CYCLE COUNT FOR 1000 XFERS
006462  012777  020510  173132          MOV     #ATEND,@BE2BA       ;SET UP ADDR REG=1ST LOCATION AFTER PROG
006470  012777  002561  173126          MOV     #2561,@BE2CR1       ;DO FUN 1-DATIP-NPR-INTR ON DONE(7)
006476  022737  000002  001702          CMP     #2,DEVCNT           ;CHECK FOR MORE THAN 2 DEVICES
006504  001444                          BEQ     1$                  ;IF NOT, GO CHECK RESULTS

006506  012777  007210  173156          MOV     #S3,@BE3VEC         ;SET UP DEVICE 3 INTR VECTOR
006514  012777  000340  173152          MOV     #PR7,@BE3PSW        ;SET UP PSW VECTOR
006522  012777  176776  173104          MOV     #-1002,@BE3CC       ;SET UP CYCLE COUNT
006530  012777  020510  173100          MOV     #ATEND,@BE3BA       ;SET UP ADDR REG
006536  012777  004005  173074          MOV     #4005,@BE3CR1       ;DO FUN 2-DATI-BR5
006544  022737  000003  001702          CMP     #3,DEVCNT           ;CHECK FOR MORE THAN 3 DEVICES
```

```
006552  001421                                  BEQ     1$                  ;IF NOT, GO CHECK RESULTS

006554  005037  001172                          CLR     $REG4               ;USE REG4 TO COUNT INTRS
006560  012777  007226  173110                  MOV     #S4,@BE4VEC         ;SET UP DEVICE 4 INTR VECTOR
006566  012777  000340  173104                  MOV     #PR7,@BE4PSW        ;SET UP PSW VECTOR
006574  012777  175000  173046                  MOV     #-3000,@BE4CC       ;SET UP CYCLE COUNT
006602  052777  040000  173050                  BIS     #BIT14,@BE4CR2      ;SET TIME DELAY BIT OF DEVICE 4
006610  012777  000021  173036                  MOV     #21,@BE4CR1         ;DO FUN 0-BR7
006616                                  1$:
006616  012700  001610                          MOV     #BE1CR1,R0          ;USE R0 TO POINT TO DEVICE 1'S CR1
006622  004737  010534                          JSR     PC,BKGD             ;PERFORM BACKGROUND ROUTINE

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;BE1 TRANSFER CHECKS
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

006626  012702  022510                          MOV     #ATEND+2000,R2      ;USE R2 TO POINT TO 2000 BYTES AFTER END OF PROGRAM
006632  012737  176000  001166                  MOV     #-2000,$REG2        ;SET UP $REG2 WITH EXPECTED RESULTS
006640  023702  001714                  10$:    CMP     DATA1,R2            ;CHECK FOR 4000 BYTES PAST END OF PROGRAM
006644  001417                                  BEQ     14$                 ;IF EQUAL, GO CHECK FOR ANOTHER DEVICE
006646  023722  001166                          CMP     $REG2,(R2)+         ;ELSE COMPARE EXPECTED RESULTS WITH THAT IN R2
006652  001004                                  BNE     12$                 ;IF NOT EQUAL, GO TO ERR MSSG
006654  062737  000002  001166                  ADD     #2,$REG2            ;INCREMENT REG2 AS CC WOULD
006662  000766                                  BR      10$                 ;AND GO BACK TO CHECK NEXT LOC
006664                                  12$:
006664  014237  001164                          MOV     -(R2),$REG1         ;MOVE R2 TO REG 5 FOR TYPEOUT
006670  010237  001162                          MOV     R2,$REG0
006674  013737  001166  001170                  MOV     $REG2,$REG3
006702  104021                                  ERROR   21                  ;TYPE ERR MSSG
006704                                  14$:
006704  022737  000001  001702                  CMP     #1,DEVCNT           ;CHECK IF ONLY 1 DEVICE SHOULD HAVE OPERATED
006712  001470                                  BEQ     TST17               ;;<IF EQUAL, GO TO NEXT TEST>

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;BE2 TRANSFER CHECKS
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

006714  012701  020510                          MOV     #ATEND,R1           ;USE R1 TO POINT TO END OF PROG
006720  023701  001716                  5$:     CMP     DATA2,R1            ;CHECK FOR 2000 BYTES PAST END
006724  001413                                  BEQ     6$                  ;IF EQUAL, EXIT
006726  022721  052525                          CMP     #052525,(R1)+       ;ELSE COMPARE EXPECTED RESULTS WITH R1
006732  001772                                  BEQ     5$                  ;IF EQUAL, GO CHECK NEXT LOC
006734  014137  001164                          MOV     -(R1),$REG1         ;MOVE R1 TO REG1 FOR TYPEOUT
006740  010137  001162                          MOV     R1,$REG0
006744  012737  052525  001170                  MOV     #052525,$REG3
006752  104020                                  ERROR   20                  ;ELSE TYPE ERR MSSG
006754                                  6$:
006754  022737  000002  001702                  CMP     #2,DEVCNT           ;CHECK IF ONLY 2 DEVICES OPERATED
006762  001444                                  BEQ     TST17               ;;<IF EQUAL, GO TO NEXT TEST>

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
              ;BE3 CHECK ROUTINE
       ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
       ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

       006764  105777  172650          16$:    TSTB    @BE3CR1          ;CHECK IF DEVICE 3 DONE
       006770  100401                          BMI     20$              ;IF YES, CHECK NEXT DEVICE
       006772  000774                          BR      16$              ;AND GO BACK TO SEE IF DONE YET
       006774                          20$:
       006774  022777  176002  172630          CMP     #-1776,@BE3DB    ;CHECK FOR CORR VALUE IN BE3DB
       007002  001407                          BEQ     25$              ;IF EQUAL, SKIP ERR TYPE OUT
       007004  017737  172622  001164          MOV     @BE3DB,$REG1     ;MOVE FOR ERR TYPE OUT
       007012  012737  176002  001166          MOV     #-1776,$REG2
       007020  104032                          ERROR   32               ;TYPE ERR MSSG
       007022                          25$:
       007022  022737  000003  001702          CMP     #3,DEVCNT        ;CHECK IF ONLY 3 DEVICES OPERATED
       007030  001421                          BEQ     TST17            ;;<IF EQUAL, GO TO NEXT TEST>

       ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
       ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
              ;BE4 CHECK ROUTINE
       ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
       ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

       007032                          22$:
       007032  105777  172616                  TSTB    @BE4CR1          ;TEST IF DEVICE 4 IS DONE
       007036  100375                          BPL     22$              ;IF NOT, GO BACK TO DELAY RTN
       007040  022737  003000  001172          CMP     #3000,$REG4      ;CHECK IF REG4 COUNTED 3000 INTRS
       007046  001407                          BEQ     26$              ;IF EQUAL SKIP ERR TYPE OUT
       007050  013737  001172  001164          MOV     $REG4,$REG1      ;MOVE FOR TYPE OUT
       007056  012737  003000  001166          MOV     #3000,$REG2
       007064  104033                          ERROR   33               ;TYPE ERR MSSG
       007066                          26$:
       007066  042777  040000  172564          BIC     #BIT14,@BE4CR2   ;ELSE CLEAR TIME DELAY BIT
                                       ;;**************************************************************
                                       ;*TEST 17        DUMMY END OF PROGRAM
                                       ;;**************************************************************
       007074  000004                  TST17:  SCOPE
       007076  012737  000001  001212          MOV     #1,$TIMES        ;;DO 1 ITERATION

       007104  017700  172036                  MOV     @$TKB,R0         ;MOVE READ BUFF CONTENTS TO R0
       007110  022700  000210                  CMP     #210,R0          ;DOES THE VALUE = ''^H'' ?
       007114  001001                          BNE     10$              ;IF NOT GO TO 10$
       007116  000000                          HALT                     ;ELSE HALT THE PROGRAM
       007120                          10$:
       007120  000137  015206                  JMP     $EOP
                                       ;;**************************************************************
                                       ;SET UP DEVICE INTR VECTOR
                                       ;;**************************************************************
       007124                          S1:

       007124  017737  172456  001714          MOV     @BE1BA,DATA1     ;MOVE ADDR IN BE1BA TO DATA1 AND
       007132  162737  000002  001714          SUB     #2,DATA1         ;SUB 2 TO GET ACTUAL ADDR
       007140  012737  000001  001206          MOV     #1,$TMP4         ;SET INDICATOR FOR DEVICE 1
       007146  005777  172436                  TST     @BE1CR1          ;TEST FOR ERROR
       007152  100041                          BPL     EXS              ;IF PLUS, EXIT
       007154  000434                          BR      CHEX             ;ELSE FIND CAUSE OF INTR
```

```
     007156                          S2:
     007156  017737  172440  001716            MOV     @BE2BA,DATA2       ;MOVE ADDR IN BE2BA TO DATA2 AND
     007164  162737  000002  001716            SUB     #2,DATA2           ;SUB 2 TO GET ACTUAL ADDR
     007172  012737  000002  001206            MOV     #2,$TMP4           ;SET INDICATOR FOR DEVICE 2
     007200  005777  172420                    TST     @BE2CR1            ;TEST FOR ERROR
     007204  100024                            BPL     EXS                ;IF PLUS EXIT
     007206  000417                            BR      CHEX               ;ELSE FIND CAUSE OF INTR
```

```
 1420  007210                           S3:
 1421  007210  012737  000003  001206             MOV      #3,$TMP4        ;SET INDICATOR FOR DEVICE 3
 1422  007216  005777  172416                     TST      @BE3CR1         ;TEST FOR ERROR
 1423  007222  100015                             BPL      EXS             ;IF PLUS, EXIT
 1424  007224  000410                     BR      CHEX              ;ELSE FIND CAUSE OF INTR
 1425  007226                     S4:
 1426  007226  005237  001172             INC     $REG4             ;COUNT DEVICE 4'S INTRS
```

```
007232  012737  000004  001206          MOV     #4,$TMP4            ;SET INDICATOR FOR DEVICE 4
007240  005777  172410                  TST     @BE4CR1             ;TEST FOR ERROR
007244  100004                          BPL     EXS                 ;IF PLUS, EXIT
007246                          CHEX:
007246  011637  001174                  MOV     (SP),$REG5          ;FOR TYPEOUT OF PC
007252  004737  010166                  JSR     PC,ERRTN            ;ELSE FIND CAUSE OF INTR
007256                          EXS:
007256  000002                          RTI
                                ;;*****************************************************************
                                ;;*****************************************************************
007260                          MULT1:
007260  012702  020510                  MOV     #ATEND,R2           ;R2 = END OF PROG
007264  012705  005000                  MOV     #5000,R5            ;R5 = THE # OF DATA WORDS
007270  004737  010610                  JSR     PC,DOUP             ;CREATE THOSE WORDS IN BUFFER MEMORY
007274  004737  010722                  JSR     PC,TSTOVR           ;SET UP PATTERN IN MEMORY BUFFER AREA
007300  012777  020510  172300          MOV     #ATEND,@BE1BA       ;SET REG ADDR= 1ST LOCATION AFTER END OF PROGRAM
007306  012777  176000  172270          MOV     #-2000,@BE1CC       ;SET CYCLE COUNT FOR 2000 XFERS
007314  012777  006202  172340          MOV     #SERV1,@BE1VEC      ;SET UP DEVICE INTR VECTOR
007322  012777  000340  172334          MOV     #PR7,@BE1PSW        ;SET UP DEVICE PSW VECTOR
007330  052777  040000  172256          BIS     #BIT14,@BE1CR2      ;SET BIT 14 FOR TIME DELAY ENABLE
007336  012777  042560  172244          MOV     #42560,@BE1CR1      ;DO DATIP/DATOB-FUN 1-NPR-INTR ON DONE(7)
007344  122737  000001  001702          CMPB    #1,DEVCNT           ;IF MORE THAN 1 DEVICE, LOAD THEIR REGISTERS
007352  001474                          BEQ     6$                  ;OTHERWISE BEGIN TESTING
007354                          3$:
007354  012777  006226  172304          MOV     #SERV2,@BE2VEC      ;SET UP DEVICE 2 INTR VECTOR
007362  012777  000340  172300          MOV     #PR7,@BE2PSW        ;SET UP DEVICE 2 PSW VECTOR
007370  012777  020510  172224          MOV     #ATEND,@BE2BA       ;SET UP ADDR REG FOR SAME LOCATIONS AS DEVICE 1
007376  012777  177000  172214          MOV     #-1000,@BE2CC       ;SET CYCLE COUNT FOR A 1000 XFERS
007404  012777  024510  172212          MOV     #24510,@BE2CR1      ;DO DATIP/NO ROTATE-FUN 2-BR6-INTR ON DONE(6)
007412  122737  000002  001702          CMPB    #2,DEVCNT           ;IF MORE THAN 2 DEVICES, LOAD THER REGISTERS
007420  001451                          BEQ     6$                  ;OTHERWISE BEGIN TESTING
007422                          4$:
007422  012777  000340  172244          MOV     #PR7,@BE3PSW        ;SET UP DEVICE 3 PSW VECTOR
007430  010377  172176                  MOV     R3,@BE3DB           ;MOVE PATTERN IN R3 TO DEVICE DATA REG
007434  012777  006252  172230          MOV     #SERV3,@BE3VEC      ;SET UP DEVICE INTR VECTOR
007442  012777  022510  172166          MOV     #ATEND+2000,@BE3BA  ;SET UP ADDR REG
007450  012777  177000  172156          MOV     #-1000,@BE3CC       ;SET UP FOR 1000 XFERS
007456  052777  040000  172160          BIS     #BIT14,@BE3CR2      ;SET BIT 14 FOR TIME DELAY ENABLE
007464  012777  003160  172146          MOV     #3160,@BE3CR1       ;DO DATO-FUN 1-FROM DB-NPR-INTR ON DONE(7)
007472  122737  000003  001702          CMPB    #3,DEVCNT           ;IF A 4TH DEVICE, GO AND LOAD REGISTERS
007500  001421                          BEQ     6$                  ;OTHERWISE BEGIN TESTING
007502                          5$:
007502  010477  172140                  MOV     R4,@BE4DB           ;MOVE PATTERN IN R4 TO DEVICE DATA REG
007506  012777  000340  172164          MOV     #PR7,@BE4PSW        ;SET UP DEVICE 4 PSW VECTOR
007514  012777  006276  172154          MOV     #SERV4,@BE4VEC      ;SET UP DEVICE 4 INTR VECTOR
007522  012777  024510  172122          MOV     #ATEND+4000,@BE4BA  ;SET UP ADDR REG
007530  012777  177000  172112          MOV     #-1000,@BE4CC       ;SET CYCLE COUNT FOR 1000 XFERS
007536  012777  003104  172110          MOV     #3104,@BE4CR1       ;DO DATO-FUN 1-BR5-INTR ON DONE
007544                          6$:
007544  005277  172030                  INC     @SIMLGO             ;START DEVICES SIMULTANEOUSLY

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;BACKGROUND ROUTINE FOR MULTITRANSFERS I

007550  012737  000001  001164          MOV     #1,$REG1            ;MOVE 1 TO TEMPORARY REG
```

```
007556  012701  001164                     MOV     #$REG1,R1         ;SET UP R1 AS POINTER
007562  005121                     7$:     COM     (R1)+             ;COMPLEMENT TEMP REG
007564  006041                             ROR     -(R1)             ;ROTATE CONTENTS RIGHT
007566  123737  001702  001166             CMPB    DEVCNT,$REG2      ;CHECK IF ALL DEVICES ARE DONE
007574  101372                             BHI     7$                ;IF NOT, CONTINUE WITH BACKGROUND RTN
                                   ;;******************************************************************

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;DEVICE 1 TRANSFER CHECKS
                ;THERE ARE NO CHECKS FOR BE2
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

007576  042777  040000  172010             BIC     #BIT14,@BE1CR2    ;CLEAR TIME DELAY BIT
007604  012700  020510                     MOV     #ATEND,R0         ;START CHECKING FOR CORRECT XFERS
007610                             8$:
007610  122720  000125                     CMPB    #125,(R0)+        ;COMPARE LOWER BYTE
007614  001012                             BNE     20$               ;IF NOT EQUAL, BR TO ERR MSSG
007616  023700  001714                     CMP     DATA1,R0          ;IS THIS LAST BYTE COMPARE?
007622  001422                             BEQ     9$                ;IF SO, BR TO 9$
007624  122720  000124                     CMPB    #124,(R0)+        ;ELSE COMPARE UPPER BYTE
007630  001006                             BNE     22$               ;IF NOT EQUAL, BR TO ERR MSSG
007632  023700  001714                     CMP     DATA1,R0          ;IS THIS LAST BYTE COMPARE?
007636  001414                             BEQ     9$                ;IF SO, BR TO 9$
007640  000763                             BR      8$                ;ELSE CHECK NEXT ADDR
007642                             20$:
007642  105740                             TSTB    -(R0)             ;SUB 1 TO GET ERR ADDR
007644  000401                             BR      24$               ;GO DO MOVES FOR ERR MSSG
007646                             22$:
007646  005740                             TST     -(R0)             ;SUB 2 TO GET ERR ADDR
007650                             24$:
007650  011037  001164                     MOV     (R0),$REG1        ;MOVES ARE FOR ERR MSSG
007654  010037  001162                     MOV     R0,$REG0
007660  012737  052125  001170             MOV     #052125,$REG3
007666  104015                             ERROR   15                ;ELSE TYPE ERR MSSG
007670                             9$:
007670  022737  000001  001702             CMP     #1,DEVCNT         ;IF ONLY ONE DEVICE
007676  001447                     25$:    BEQ     13$               ;IF NO MORE DEVICES,EXIT RTN
007700  122737  000002  001702             CMPB    #2,DEVCNT         ;CHECK FOR MORE THAN 2 DEVICES
007706  001773                             BEQ     25$               ;IF NOT, EXIT TEST

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;BE3 TRANSFER CHECKS
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

007710  042777  040000  171726             BIC     #BIT14,@BE3CR2    ;CLEAR TIME DELAY BIT OF DEVICE 3
007716  012700  022510                     MOV     #ATEND+2000,R0    ;CHECK NEXT 2000 LOCATIONS
007722  023700  001720             10$:    CMP     DATA3,R0          ;CHECK FOR 1000 XFERS
007726  001411                             BEQ     11$               ;IF SO, CHECK NEXT BLOCK
007730  020320                             CMP     R3,(R0)+          ;TEST FOR CORRECT PATTERNS
007732  001773                             BEQ     10$               ;IF NO ERR, CHECK ANOTHER LOC
007734  010337  001170                     MOV     R3,$REG3          ;THE MOVE IS FOR TYPEOUT REASONS
007740  014037  001164                     MOV     -(R0),$REG1
```

```
007744  010037  001162                          MOV     R0,$REG0
007750  104016                                  ERROR   16                      ;ELSE TYPE ERR MSSG
007752                                  11$:
007752  122737  000003  001702                  CMPB    #3,DEVCNT               ;CHECK FOR MORE THAN 3 DEVICES
007760  001416                                  BEQ     13$                     ;IF NO MORE DEVICES GO DO NEXT PATTERN

;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;BE4 TRANSFER CHECKS
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

007762  012700  024510                          MOV     #ATEND+4000,R0          ;CHECK NEXT BLOCK OF 2000 LOCATIONS
007766  023700  001722                  12$:    CMP     DATA4,R0                ;CHECK FOR 1000 XFERS
007772  001411                                  BEQ     13$                     ;IF SO, CHANGE DATA PATTERNS & START OVER AGAIN
007774  020420                                  CMP     R4,(R0)+                ;ELSE CHECK FOR CORRECT PATTERNS
007776  001773                                  BEQ     12$                     ;IF NO ERR, CHECK ANOTHER LOCATION
010000  010437  001170                          MOV     R4,$REG3                ;THE MOVE IS FOR TYPEOUT REASONS
010004  014037  001164                          MOV     -(R0),$REG1
010010  010037  001162                          MOV     R0,$REG0
010014  104017                                  ERROR   17                      ;ELSE TYPE ERR MSSG
010016                                  13$:
010016  000207                                  RTS     PC
                                        ;//////////////////////////////////////////////////////////////////////////
                                        ;//////////////////////////////////////////////////////////////////////////
                                        ;//////////////////////////////////////////////////////////////////////////
010020                                  STVEC:
010020  005000                                  CLR     R0                      ;R0 IS COUNTER
010022  012701  010056                          MOV     #Q1,R1                  ;R1 IS ADDR OF INTR RTN
010026  012702  000510                          MOV     #510,R2                 ;R2 IS DEV INTR VEC ADDR
010032                                  5$:
010032  010122                                  MOV     R1,(R2)+                ;LOAD INTR RTN INTO INTR VEC
010034  012722  000340                          MOV     #PR7,(R2)+              ;LOAD INTR PSW
010040  062701  000006                          ADD     #6,R1                   ;GET NEXT INTR RTN
010044  005200                                  INC     R0                      ;INC COUNTER
010046  020037  001702                          CMP     R0,DEVCNT               ;ARE ALL AVAILABLE VECS LOADED?
010052  001367                                  BNE     5$                      ;IF NOT, GO AND LOAD NEXT ONE
010054  000207                                  RTS     PC                      ;ELSE EXIT
                                        ;;********************************************************************
                                        ;;********************************************************************

                                        ; THE FOLLOWING INTR ROUTINES SHOULD HANDLE ANY ERRORS
                                        ; WHICH MIGHT OCCUR BEFORE THE PROGRAM HAS A CHANCE TO
                                        ; PROPERLY CHECK OUT ALL DEVICES AND SET UP THEIR INTR VECTORS


010056                                  Q1:
010056  012737  000001  001206                  MOV     #1,$TMP4                ;INDICATOR FOR DEV 1
010064  000413                                  BR      XQ
010066                                  Q2:
010066  012737  000002  001206                  MOV     #2,$TMP4                ;INDICATOR FOR DEV 2
010074  000407                                  BR      XQ
010076                                  Q3:
010076  012737  000003  001206                  MOV     #3,$TMP4                ;INDICATOR FOR DEV 3
010104  000403                                  BR      XQ
010106                                  Q4:
```

```
010106  012737  000004  001206          MOV     #4,$TMP4              ;INDICATOR FOR DEV 4
010114                          XQ:
010114  004737  010166                  JSR     PC,ERRTN              ;GO TO ERR ROUTINE
010120  000002                          RTI
                                ;;******************************************************
                                ;;******************************************************
010122                          PWRFAL:
010122  010146                          MOV     R1,-(SP)              ;SAVE CONTENTS OF R1
010124  010046                          MOV     R0,-(SP)              ;SAVE CONTENTS OF R0
010126  012700  001614                  MOV     #BE1CR2,R0            ;R0 POINTS TO DEV 1 CR2 ADDR
010132  005001                          CLR     R1                    ;CLEAR R1
010134                          5$:
010134  042770  000020  000000          BIC     #BIT04,@(R0)          ;CLR BIT 4 OF CURRENT CR2
010142  062700  000014                  ADD     #14,R0                ;ADD 14 TO POINT TO NEXT CR2
010146  005201                          INC     R1                    ;COUNT THE NUMBER OF DEVS
010150  020137  001702                  CMP     R1,DEVCNT             ;REACHED MAX # ON BUS?
010154  103767                          BLO     5$                    ;IF NOT, CLR NEXT CR2
010156  012600                          MOV     (SP)+,R0              ;ELSE RESTORE R0
010160  012601                          MOV     (SP)+,R1              ;AND R1
010162  000137  020332                  JMP     $PWRDN                ;THEN DO REGULAR PWR DOWN RTN
                                ;;******************************************************
                                ;;******************************************************
010166                          ERRTN:
010166  104410                          SAVREG                        ;SAVE REGISTERS
010170  012700  001600                  MOV     #BE1CR2-14,R0 ;INITIALIZE R0
010174  105005                          CLRB    R5                    ;CLEAR DEVICE COUNTER
010176                          1$:
010176  105205                          INCB    R5                    ;ADD 1 TO COUNTER
010200  062700  000014                  ADD     #14,R0                ;SET R0=ADDR OF CR2 OF NEXT DEVICE
010204  120537  001206                  CMPB    R5,$TMP4              ;IF COUNTER NOT EQUAL TO INDICATOR
010210  001372                          BNE     1$                    ;ADD 1 TO COUNTER & CHECK AGAIN
010212                          CHKERR:
010212  105770  000000                  TSTB    @(R0)                 ;CHECK FOR NO NOSACK TIMEOUT
010216  100001                          BPL     1$                    ;IF NOT, SEE IF THERE ARE ANY ERRS
010220  104022                          ERROR   22                    ;TYPE ERR MESSG FOR NO NOSACK
010222                          1$:
010222  032770  007540  000000          BIT     #7540,@(R0)           ;CHECK FOR OTHER ERRORS
010230  001436                          BEQ     LEEV                  ;IF NO ERRORS,EXIT
010232  032770  004000  000000          BIT     #BIT11,@(R0)          ;CHECK FOR NO SSYN ON INTR
010240  001401                          BEQ     10$                   ;IF NOT SET, CHECK FOR NEXT ERR
010242  104023                          ERROR   23                    ;TYPE ERR MSSG FOR NO SSYN ON INTR
010244                          10$:
010244  132770  000040  000000          BITB    #BIT05,@(R0)          ;CHECK FOR WRONG GRANT ERR
010252  001401                          BEQ     2$                    ;IF NOT, CHECK BIT 6
010254  104024                          ERROR   24                    ;ELSE TYPE ERR MESSG FOR WRONG GRANT
010256                          2$:
010256  132770  000100  000000          BITB    #BIT06,@(R0)          ;CHECK FOR BUS LATE ERR
010264  001401                          BEQ     3$                    ;IF NOT, CHECK BIT 8
010266  104025                          ERROR   25                    ;TYPE ERR MSSG FOR BUS LATE
010270                          3$:
010270  032770  000400  000000          BIT     #BIT08,@(R0)          ;CHECK FOR NO SSYN ERR
010276  001401                          BEQ     4$                    ;IF NOT, CHECK BIT 9
010300  104026                          ERROR   26                    ;TYPE ERR MSSG FOR NO SSYN
010302                          4$:
010302  032770  001000  000000          BIT     #BIT09,@(R0)          ;CHECK FOR WRONG ADDR ERR
010310  001401                          BEQ     5$                    ;IF NOT, CHECK BIT 10
```

```
010312  104027                          ERROR   27              ;TYPE ERR MSSG FOR WRONG ADDR
010314                          5$:
010314  032770  002000  000000          BIT     #BIT10,@(R0)    ;CHECK FOR NO GRANT ERR
010322  001401                          BEQ     LEEV            ;IF NOT, EXIT
010324  104030                          ERROR   30              ;TYPE ERR MSSG FOR NO GRANT
010326                          LEEV:
010326  162700  000002                  SUB     #2,R0           ;POINT TO DEVICE CLEAR REG
010332  005070  000000                  CLR     @(R0)           ;CLEAR ALL ERRORS
010336  104411                          RESREG                  ;RESTORE REGISTERS
010340  000207                          RTS     PC              ;EXIT
                                ;;************************************************************
                                ;;************************************************************
                                ;;************************************************************
010342                          THRU0:
010342  011637  001166                  MOV     (SP),$REG2      ;MOVE FOR ERR TYPE OUT
010346  162737  000002  001166          SUB     #2,$REG2        ;SUB 2 FOR ACTUAL ADDR
010354  104035                          ERROR   35              ;TYPE ERR MSSG
010356  000002                          RTI
                                ;;************************************************************
                                ;;************************************************************
010360                          ROTATE:
010360  032703  000001                  BIT     #BIT00,R3       ;IS LSB A 1 OR 0 ?
010364  001402                          BEQ     5$              ;IF 0, GO TO 5$
010366  000261                          SEC                     ;ELSE SET C BIT OF COND CODES
010370  000401                          BR      10$             ;AND GO ROTATE
010372                          5$:
010372  000241                          CLC                     ;CLEAR C BIT OF COND CODES
010374                          10$:
010374  006003                          ROR     R3              ;ROTATE R3
010376  032704  000001                  BIT     #BIT00,R4       ;IS LSB A 1 OR 0 ?
010402  001402                          BEQ     15$             ;IF 0, GO TO 15$
010404  000261                          SEC                     ;ELSE SET C BIT OF COND CODES
010406  000401                          BR      20$             ;AND GO ROTATE
010410                          15$:
010410  000241                          CLC                     ;CLEAR C BIT OF COND CODES
010412                          20$:
010412  006004                          ROR     R4              ;ROTATE R4
010414  000207                          RTS     PC
                                ;;************************************************************
                                ;;************************************************************
010416                          NOG:
010416                          2$:
010416  010037  177776                  MOV     R0,PSW          ;SET UP PROCESSOR STATUS
010422  012777  020510  171156          MOV     #ATEND,@BE1BA   ;SET UP ADDR REG
010430  012777  177777  171146          MOV     #-1,@BE1CC      ;SET UP CYCLE COUNT FOR 1 CYCLE
010436  010177  171146                  MOV     R1,@BE1CR1      ;DO FUN 1 ON BR LEVELS IN R1
010442  000240                          NOP                     ;WAIT FOR DEVICE TO ATTEMPT TO  DO XFER
010444  022777  177777  171132          CMP     #-1,@BE1CC      ;SEE IF DEVICE OPERATED
010452  001005                          BNE     4$              ;IF IT DID,GO TYPE ERR MSSG
010454  106201                          ASRB    R1              ;SHIFT BYTE RIGHT TO LOWER BRR0
010456  122701  000001                  CMPB    #1,R1           ;IF BYTE IS NOT EQUAL TO 1
010462  001355                          BNE     2$              ;GO TO 2$
010464  000402                          BR      EXNOG           ;EXIT
010466                          4$:
010466  004737  010474                  JSR     PC,ERRS
010472  000207                  EXNOG:  RTS     PC              ;EXIT SUB RTN
```

```
                                        ;;*******************************************************************
 010474                                 ERRS:
 010474  017737  171102  001162                 MOV     @BE1DB,$REG0      ;MOVES ARE FOR TYPEOUTS
 010502  017737  171076  001164                 MOV     @BE1CC,$REG1
 010510  017737  171072  001166                 MOV     @BE1BA,$REG2
 010516  017737  171066  001170                 MOV     @BE1CR1,$REG3
 010524  010037  001172                         MOV     R0,$REG4
 010530  104002                                 ERROR   2
 010532  000207                                 RTS     PC                ;EXIT ERROR RTN
                                        ;;*******************************************************************
                                        ;;*******************************************************************
 010534                                 BKGD:
 010534  012737  031463  001164                 MOV     #031463,$REG1     ;START OF BACKGROUND ROUTINE
 010542  012701  001165                         MOV     #$REG1+1,R1       ;USE R1 TO POINT TO TEST PATTERN
 010546  105441                         1$:     NEGB    -(R1)             ;DECREMENT LOC AND NEGATE BYTE=(031715)
 010550  105421                                 NEGB    (R1)+             ;NEGATE BYTE THEN INCREMENT LOC=(031463)
 010552  105421                                 NEGB    (R1)+             ;NEGATE BYTE THEN INCREMENT LOC=(146463)
 010554  105770  000000                         TSTB    @(R0)             ;TEST FOR DONE BIT OF DEVICE IN R0
 010560  100402                                 BMI     2$                ;IF DONE, GO CHECK RESULTS
 010562  105441                                 NEGB    -(R1)             ;ELSE DECREMENT LOC AND NEGATE BYTE=(031463)
 010564  000770                                 BR      1$                ;CONTINUE WITH BACKGROUND
 010566                                 2$:

 010566  005741                                 TST     -(R1)             ;BRING POINTER DOWN TO REG1
 010570  022711  146463                         CMP     #146463,(R1)      ;COMPARE EXPECTED PATTERN WITH THAT IN R1
 010574  001404                                 BEQ     BKEX              ;IF EQUAL, EXIT THIS RTN
 010576  012737  146463  001166                 MOV     #146463,$REG2     ;MOVE FOR TYPE OUT
 010604  104031                                 ERROR   31                ;ELSE TYPE ERR MSSG
 010606  000207                         BKEX:   RTS     PC
                                        ;;*******************************************************************
                                        ;;*******************************************************************
 010610                                 DOUP:
 010610  012701  000001                         MOV     #1,R1             ;INIT R1 TO 1
 010614                                 5$:
 010614  010122                                 MOV     R1,(R2)+          ;MOVE CONTENTS OF R1 TO AREA IN R2
 010616  005201                                 INC     R1                ;ADD 1 TO R1
 010620  020105                                 CMP     R1,R5             ;IS # OF MOVES = TO # IN R5?
 010622  101774                                 BLOS    5$                ;IF NOT, DO ANOTHER MOVE
 010624  000207                                 RTS     PC                ;ELSE EXIT
                                        ;;*******************************************************************
                                        ;;*******************************************************************

 010626                                 CNTR:
 010626  012737  000001  001172                 MOV     #1,$REG4          ;INITIALIZE COUNTER REG
 010634  062737  000001  001172         1$:     ADD     #1,$REG4          ;ADD 1 TO IT
 010642  022737  000106  001172                 CMP     #70.,$REG4        ;DELAY AT LEAST 41 US
 010650  001371                                 BNE     1$                ;IF NOT, GO BACK AND ADD 1 TO REG4
 010652  000207                                 RTS     PC                ;EXIT
                                        ;;*******************************************************************
                                        ;;*******************************************************************
 010654                                 ADLI:
 010654  012700  040000                         MOV     #40000,R0         ;USE R0 TO SET BIT 14
 010660                                 1$:
 010660  012777  177777  170716                 MOV     #-1,@BE1CC        ;SET CYCLE COUNT = 1 XFER
 010666  010077  170714                         MOV     R0,@BE1BA         ;SET ADDR AS SPECIFIED IN R0
 010672  012777  002041  170710                 MOV     #2041,@BE1CR1     ;DO DATI-FUN 1-NPR
```

```
010700  004737  010626                  JSR     PC,CNTR             ;ALLOW TIME FOR RDY BIT TO SET
010704  022700  100000                  CMP     #100000,R0          ;CHECK IF BIT 15 OF R0 IS SET
010710  001403                          BEQ     EXAD                ;IF SET, GO SET NEXT ADDR LINE
010712  012700  100000                  MOV     #100000,R0          ;ELSE, NOW SET BIT 15 OF R0
010716  000760                          BR      1$                  ;GO BACK AND CHECK THAT ADDR
010720                          EXAD:
010720  000207                          RTS     PC                  ;EXIT SUB ROUTINE
                                ;;*****************************************************************
                                ;;*****************************************************************
010722                          TSTOVR:
010722  012737  000140  177776          MOV     #PR3,PSW            ;PS=3
010730  005037  001166                  CLR     $REG2               ;CLEAR REG FOR INTR ON DONE COUNTER
010734  012700  020510                  MOV     #ATEND,R0           ;SET UP R0 AS POINTER
010740  012720  125252          1$:     MOV     #125252,(R0)+       ;MOVE DATA PATTERN TO AVAILABLE MEMORY
010744  022700  022510                  CMP     #ATEND+2000,R0 ;CHECK FOR A 2000 MOVES
010750  001373                          BNE     1$                  ;IF NOT, GO BACK AND MOVE AGAIN
010752  000207                          RTS     PC                  ;EXIT
                                ;;*****************************************************************
                                ;;*****************************************************************

010754  005015  005015  047125  QNO:    .ASCIZ  <15><12><15><12>\UNIBUS SYSTEMS EXERCISER DIAGNOSTIC - CZKUA-E \
011037     015  042012  053105  FOR4:   .ASCIZ  <15><12>\DEV 4 MUST HAVE TIME DELAY SET @ 100 US OR LATENCY ERR MAY OCCU
011144  050103  020125  051124  EM1:    .ASCIZ \CPU TRAPPED THRU LOCATION 4 -TIMEOUT \
011212  040440  042104  020122  DH1:    .ASCIZ \ ADDR   $ERRPC #ERR/TST#\
011243     103  052520  044440  EM2:    .ASCII  \CPU ISSUED A BUS GRANT WITH PSW = 7\
011306  042504  020126  020061          .ASCIZ  \DEV 1 SHOULD NOT HAVE BECOME BUS MASTER\
011356  042502  042061  020102  DH2:    .ASCIZ  \BE1DB   BE1CC   BE1BA   BE1CR1   PSW   $ERRPC #ERR/TST#\
011446  050103  020125  044504  EM3:    .ASCIZ \CPU DID NOT ISSUE BUS NPG\
011500  042502  041461  030522  DH3:    .ASCIZ  \BE1CR1   BE1CC  FM-PS   TO-PS   $ERRPC #ERR/TST#\
011561     103  052520  042040  EM4:    .ASCIZ  \CPU DID NOT ISSUE BUS GRANT 7\
011617     103  052520  042040  EM5:    .ASCIZ  \CPU DID NOT ISSUE BUS GRANT 6\
011655     103  052520  042040  EM6:    .ASCIZ  \CPU DID NOT ISSUE BUS GRANT 5\
011713     103  052520  042040  EM7:    .ASCIZ  \CPU DID NOT ISSUE BUS GRANT 4\
011751     117  042516  047440  EM10:   .ASCIZ  \ONE OR MORE DEVICES DID NOT INTERRUPT\
012017     124  044510  020123  DH10:   .ASCII  \THIS IS THE ORDER IN WHICH THEY INTERRUPTED\<15><12>
012074  020040  051461  020124          .ASCIZ  \  1ST     2ND     3RD     4TH   $ERRPC #ERR/TST#\
012155     102  051525  040440  EM11:   .ASCIZ  \BUS ADDRESS LINES <A17:A14> DID NOT FUNCTION PROPERLY\
012243     102  030505  051103  DH11:   .ASCIZ  \BE1CR1  BE1CR2   BE1BA  $ERRPC #ERR/TST#\
012314  050103  020125  047516  EM12:   .ASCIZ  \CPU NO SACK TIMEOUT LOGIC FAILED(TO NEGATE BUS GRANT)\
012402  042502  041461  030522  DH12:   .ASCIZ  \BE1CR1  BE1CR2  $ERRPC #ERR/TST#\
012443     103  052520  042040  EM13:   .ASCIZ  \CPU DID NOT PROPERLY EXECUTE AN ACLO/DCLO SEQUENCE\
012526  042444  051122  041520  DH13:   .ASCIZ  \$ERRPC #ERR/TST#\
012547     103  052520  042040  EM14:   .ASCIZ  \CPU DID NOT TRAP FROM BUS PARITY ERROR PA/PB = 0/1\
012632  042504  020126  020061  EM15:   .ASCII  \DEV 1 DID DATIP WITH ROL ON DATOB TO MEMORY\<15><12>
012707     124  042510  052040          .ASCIZ  \THE TRANSFER TO THE FOLLOWING LOCATION WAS  INCORRECT\
012775     115  046505  051117  DH15:   .ASCII  \MEMORY  ACTUAL  CORRECT\<15><12>
013026  046040  041517  020040          .ASCIZ  \ LOC      DATA     DATA   $ERRPC #ERR/TST# $ICNT #\
013107     104  053105  041511  EM16:   .ASCIZ  \DEVICE 3'S DATO TO MEMORY DID NOT EQUAL PATTERN IN R3\
013175     104  053105  041511  EM17:   .ASCIZ  \DEVICE 4'S DATO TO MEMORY DID NOT EQUAL PATTERN IN R4\
013263     104  053105  041511  EM20:   .ASCIZ \DEVICE 1 DOING FUN 1-NPR-DATIP; INCORRECT PATTERN IN MEMORY\
013357     104  053105  041511  EM21:   .ASCIZ \DEVICE 2 DOING FUN 2-NPR-DATOB; INCORRECT PATTERN IN MEMORY\
013453     102  052111  033440  EM22:   .ASCIZ  \BIT 7 OF CR2 SET-CPU DID NOT TIME OUT WITH SACK INHIBITED\
013545     104  053105  021440  DH22:   .ASCIZ  \DEV #      PC    $ERRPC  #ERR/TST#\
013607     102  052111  030440  EM23:   .ASCIZ  \BIT 11 OF CR2 SET-NO SSYN ON INTR SIGNAL\
013660  044502  020124  020065  EM24:   .ASCIZ  \BIT 5 OF CR2 SET-RECEIVED WRONG GRANT\
013726  044502  020124  020066  EM25:   .ASCIZ  \BIT 6 OF CR2 SET-BUS LATE\
        013760  044502  020124  020070  EM26:   .ASCIZ  \BIT 8 OF CR2 SET-NO SSYN OCCURRED\
        014022  044502  020124  020071  EM27:   .ASCIZ  \BIT 9 OF  CR2 SET-WRONG ADDRESS ON BUS\
        014071     102  052111  030440  EM30:   .ASCIZ  \BIT 10 OF CR2 SET-DEVICE RECEIVED OTHER THAN ONE GRANT\
        014160  045502  047107  020104  EM31:   .ASCII  \BKGND ROUTINE INSTRUCTIONS OF NEGB'S WERE NOT DONE\<15><12>
        014244  047503  051122  041505          .ASCIZ  \CORRECTLY TO $REG1 DURING MULTITRANSFERS II\
        014320  041501  052524  046101  DH31:   .ASCII  \ACTUAL   CORR'T   \<15><12>
        014342  040504  040524  020040          .ASCIZ  \DATA     DATA     $ERRPC #ERR/TST# $ICNT #\
        014413     104  053105  031440  EM32:   .ASCIZ  \DEV 3 DID DATI BUT HAS INCORRECT VALUES IN DATA REG\
        014477     104  053105  032040  EM33:   .ASCIZ  \DEV 4 DID NOT INTR THE CORRECT # OF TIMES\
        014551     114  051501  020124  EM34:   .ASCII  \LAST DATIP TO DEVICE 1 DB WAS INCORRECT- EITHER DEVICE DID\<15><12>
        014645     116  052117  053440          .ASCIZ  \NOT WORK OR BUFFER AREA WAS NOT SET UP PROPERLY\
        014725     015  041412  052520  EM35:   .ASCIZ  <15><12>\CPU TRAPPED THRU LOC 0 TO CATCH IMPROPERLY LOADED VECTORS\
                015022                          .EVEN
        015022  001166  001116  001102  DT1:    .WORD   $REG2,$ERRPC,$TSTNM,0
        015032  001162  001164  001166  DT2:    .WORD   $REG0,$REG1,$REG2,$REG3,$REG4,$ERRPC,$TSTNM,0
        015052  001162  001164  001166  DT3:    .WORD   $REG0,$REG1,$REG2,$REG3,$ERRPC,$TSTNM,0
        015070  001164  001166  001170  DT10:   .WORD   $REG1,$REG2,$REG3,$REG4,$ERRPC,$TSTNM,0
        015106  001164  001166  001170  DT11:   .WORD   $REG1,$REG2,$REG3,$ERRPC,$TSTNM,0
        015122  001162  001164  001116  DT12:   .WORD   $REG0,$REG1,$ERRPC,$TSTNM,0
        015134  001116  001102  000000  DT13:   .WORD   $ERRPC,$TSTNM,0
        015142  001162  001164  001170  DT15:   .WORD   $REG0,$REG1,$REG3,$ERRPC,$TSTNM,$ICNT,0
        015160  001206  001174  001116  DT22:   .WORD   $TMP4,$REG5,$ERRPC,$TSTNM,0
        015172  001164  001166  001116  DT31:   .WORD   $REG1,$REG2,$ERRPC,$TSTNM,$ICNT,0
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        .SBTTL  END OF PASS ROUTINE

                                        ;;*****************************************************************
                                        ;;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;;*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY''
                                        ;;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                        ;;*IF THERES A MONITOR GO TO IT
                                        ;;*IF THERE ISN'T JUMP TO TST1

        015206                          $EOP:
        015206  000004                          SCOPE
        015210  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
        015214  005037  001212                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
        015220  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
        015224  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
        015232  005327                          DEC     (PC)+           ;;LOOP?
        015234  000001                  $EOPCT: .WORD   1
        015236  003063                          BGT     $DOAGN          ;;YES
        015240  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
        015242  000001                  $ENDCT: .WORD   1
        015244  015234                          $EOPCT
        015246  104401  015254                  TYPE    ,65$            ;;TYPE ASCIZ STRING
        015252  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <12><15>/END PASS #/
        015272                          64$:
        015272  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                        ;;TYPE PASS NUMBER
        015276  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
        015300  104401  015306                  TYPE    ,67$            ;;TYPE ASCIZ STRING
        015304  000421                          BR      66$             ;;GET OVER THE ASCIZ
```

```
                                         ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
015350                                   66$:
015350  013746  001112                          MOV     $ERTTL,-(SP)     ;;SAVE $ERTTL FOR TYPEOUT
                                                                          ;;TOTAL NUMBER OF ERRORS
015354  104405                                  TYPDS                    ;;GO TYPE--DECIMAL ASCII WITH SIGN
015356  104401  001223                          TYPE    ,$CRLF           ;;TYPE CARRIAGE RETURN. LINE FEED
015362  005037  001112                          CLR     $ERTTL           ;;CLEAR ERROR TOTAL
015366  013700  000042                   $GET42: MOV    @#42,R0          ;;GET MONITOR ADDRESS
015372  001405                                  BEQ     $DOAGN           ;;BRANCH IF NO MONITOR
015374  000005                                  RESET                    ;;CLEAR THE WORLD
015376  004710                           $ENDAD: JSR    PC,(R0)          ;;GO TO MONITOR
015400  000240                                  NOP                      ;;SAVE ROOM
015402  000240                                  NOP                      ;;FOR
015404  000240                                  NOP                      ;;ACT11
015406                                   $DOAGN:
015406  000137                                  JMP     @(PC)+           ;;RETURN
015410  003252                           $RTNAD: .WORD  TST1
015412     377     377     000           $ENULL: .BYTE  -1,-1,0          ;;NULL CHARACTER STRING
        015416                                  .EVEN
                                         .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
                                         ;*SW08=1          LOOP ON TEST IN SWR<5:0>
                                         ;*CALL
                                         ;*      SCOPE                  ;;SCOPE=IOT

015416                                   $SCOPE:
015416  032777  040000  163514           1$:    BIT     #BIT14,@SWR      ;;LOOP ON PRESENT TEST?
015424  001114                                  BNE     $OVER            ;;YES IF SW14=1
                                         ;#####START OF CODE FOR THE XOR TESTER#####
015426  000416                           $XTSTR: BR     6$               ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                          ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
015430  013746  000004                          MOV     @#ERRVEC,-(SP)   ;;SAVE THE CONTENTS OF THE ERROR VECTOR
015434  012737  015454  000004                  MOV     #5$,@#ERRVEC     ;;SET FOR TIMEOUT
015442  005737  177060                          TST     @#177060         ;;TIME OUT ON XOR?
015446  012637  000004                          MOV     (SP)+,@#ERRVEC   ;;RESTORE THE ERROR VECTOR
015452  000466                                  BR      $SVLAD           ;;GO TO THE NEXT TEST
015454  022626                           5$:    CMP     (SP)+,(SP)+      ;;CLEAR THE STACK AFTER A TIME OUT
015456  012637  000004                          MOV     (SP)+,@#ERRVEC   ;;RESTORE THE ERROR VECTOR
015462  000426                                  BR      7$               ;;LOOP ON THE PRESENT TEST
015464                                   6$:;#####END OF CODE FOR THE XOR TESTER#####
015464  032777  000400  163446                  BIT     #BIT08,@SWR      ;;LOOP ON SPEC. TEST?
015472  001407                                  BEQ     2$               ;;BR IF NO
015474  017746  163440                          MOV     @SWR,-(SP)       ;;SET DESIRED TEST NUM. FROM SWR
015500  042716  000300                          BIC     #$SWRMK,(SP)     ;;STRIP AWAY UNDESIRED BITS
015504  122637  001102                          CMPB    (SP)+,$TSTNM     ;;ON THE RIGHT TEST?
015510  001462                                  BEQ     $OVER            ;;BR IF YES
015512  105737  001103                   2$:    TSTB    $ERFLG           ;;HAS AN ERROR OCCURRED?
015516  001421                                  BEQ     3$               ;;BR IF NO
```

```
015520  123737  001115  001103          CMPB    $ERMAX,$ERFLG     ;;MAX. ERRORS FOR THIS TEST OCCURRED?
015526  101015                          BHI     3$                ;;BR IF NO
015530  032777  001000  163402          BIT     #BIT09,@SWR       ;;LOOP ON ERROR?
015536  001404                          BEQ     4$                ;;BR IF NO
015540  013737  001110  001106  7$:     MOV     $LPERR,$LPADR     ;;SET LOOP ADDRESS TO LAST SCOPE
015546  000443                          BR      $OVER
015550  105037  001103          4$:     CLRB    $ERFLG            ;;ZERO THE ERROR FLAG
015554  005037  001212                  CLR     $TIMES            ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
015560  000415                          BR      1$                ;;ESCAPE TO THE NEXT TEST
015562  032777  004000  163350  3$:     BIT     #BIT11,@SWR       ;;INHIBIT ITERATIONS?
015570  001011                          BNE     1$                ;;BR IF YES
015572  005737  001100                  TST     $PASS             ;;IF FIRST PASS OF PROGRAM
015576  001406                          BEQ     1$                ;;      INHIBIT ITERATIONS
015600  005237  001104                  INC     $ICNT             ;;INCREMENT ITERATION COUNT
015604  023737  001212  001104          CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
015612  002021                          BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
015614  012737  000001  001104  1$:     MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
015622  013737  015672  001212          MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
015630  105237  001102          $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
015634  011637  001106                  MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
015640  011637  001110                  MOV     (SP),$LPERR       ;;SAVE ERROR LOOP ADDRESS
015644  005037  001214                  CLR     $ESCAPE           ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
015650  112737  000001  001115          MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
015656  013777  001102  163256  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
015664  013716  001106                  MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
015670  000002                          RTI                       ;;FIXES PS
015672  000040                  $MXCNT: 40                        ;;MAX. NUMBER OF ITERATIONS
                                .SBTTL  ERROR HANDLER ROUTINE


                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*       ERROR   N         ;;ERROR=EMT AND N=ERROR ITEM NUMBER

015674                          $ERROR:
015674  105237  001103          7$:     INCB    $ERFLG            ;;SET THE ERROR FLAG
015700  001775                          BEQ     7$                ;;DON'T LET THE FLAG GO TO ZERO
015702  013777  001102  163232          MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER AND ERROR FLAG
015710  032777  002000  163222          BIT     #BIT10,@SWR       ;;BELL ON ERROR?
015716  001402                          BEQ     1$                ;;NO - SKIP
015720  104401  001216                  TYPE    ,$BELL            ;;RING BELL
015724  005237  001112          1$:     INC     $ERTTL            ;;COUNT THE NUMBER OF ERRORS
015730  011637  001116                  MOV     (SP),$ERRPC       ;;GET ADDRESS OF ERROR INSTRUCTION
015734  162737  000002  001116          SUB     #2,$ERRPC
015742  117737  163150  001114          MOVB    @$ERRPC,$ITEMB    ;;STRIP AND SAVE THE ERROR ITEM CODE
015750  032777  020000  163162          BIT     #BIT13,@SWR       ;;SKIP TYPEOUT IF SET
015756  001004                          BNE     20$               ;;SKIP TYPEOUTS
015760  004737  016042                  JSR     PC,$ERRTYP        ;;GO TO USER ERROR ROUTINE
015764  104401  001223                  TYPE    ,$CRLF
```

```
015770                                   20$:
015770  005777  163144                   2$:     TST     @SWR              ;;HALT ON ERROR
015774  100001                                   BPL     3$                ;;SKIP IF CONTINUE
015776  000000                                   HALT                      ;;HALT ON ERROR!
016000  032777  001000  163132           3$:     BIT     #BIT09,@SWR       ;;LOOP ON ERROR SWITCH SET?
016006  001402                                   BEQ     4$                ;;BR IF NO
016010  013716  001110                           MOV     $LPERR,(SP)       ;;FUDGE RETURN FOR LOOPING
016014  005737  001214                   4$:     TST     $ESCAPE           ;;CHECK FOR AN ESCAPE ADDRESS
016020  001402                                   BEQ     5$                ;;BR IF NONE
016022  013716  001214                           MOV     $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE
016026                                   5$:
016026  022737  015376  000042                   CMP     #$ENDAD,@#42      ;;ACT-11 AUTO-ACCEPT?
016034  001001                                   BNE     6$                ;;BRANCH IF NO
016036  000000                                   HALT                      ;;YES
016040                                   6$:
016040  000002                                   RTI                       ;;RETURN
                                         .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

016042                                   $ERRTYP:
016042  104401  001223                           TYPE    ,$CRLF            ;;''CARRIAGE RETURN'' & ''LINE FEED''
016046  010046                                   MOV     R0,-(SP)          ;;SAVE R0
016050  005000                                   CLR     R0                ;;PICKUP THE ITEM INDEX
016052  153700  001114                           BISB    @#$ITEMB,R0
016056  001004                                   BNE     1$                ;;IF ITEM NUMBER IS ZERO, JUST
                                                                           ;;TYPE THE PC OF THE ERROR
016060  013746  001116                           MOV     $ERRPC,-(SP)      ;;SAVE $ERRPC FOR TYPEOUT
                                                                           ;;ERROR ADDRESS
016064  104402                                   TYPOC                     ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
016066  000426                                   BR      6$                ;;GET OUT
016070  005300                           1$:     DEC     R0                ;;ADJUST THE INDEX SO THAT IT WILL
016072  006300                                   ASL     R0                ;;        WORK FOR THE ERROR TABLE
016074  006300                                   ASL     R0
016076  006300                                   ASL     R0
016100  062700  001226                           ADD     #$ERRTB,R0        ;;FORM TABLE POINTER
016104  012037  016114                           MOV     (R0)+,2$          ;;PICKUP ''ERROR MESSAGE'' POINTER
016110  001404                                   BEQ     3$                ;;SKIP TYPEOUT IF NO POINTER
016112  104401                                   TYPE                      ;;TYPE THE ''ERROR MESSAGE''
016114  000000                           2$:     .WORD   0                 ;;''ERROR MESSAGE'' POINTER GOES HERE
016116  104401  001223                           TYPE    ,$CRLF            ;;''CARRIAGE RETURN'' & ''LINE FEED''
016122  012037  016132                   3$:     MOV     (R0)+,4$          ;;PICKUP ''DATA HEADER'' POINTER
016126  001404                                   BEQ     5$                ;;SKIP TYPEOUT IF 0
016130  104401                                   TYPE                      ;;TYPE THE ''DATA HEADER''
016132  000000                           4$:     .WORD   0                 ;;''DATA HEADER'' POINTER GOES HERE
016134  104401  001223                           TYPE    ,$CRLF            ;;''CARRIAGE RETURN'' & ''LINE FEED''
016140  011000                           5$:     MOV     (R0),R0           ;;PICKUP ''DATA TABLE'' POINTER
016142  001004                                   BNE     7$                ;;GO TYPE THE DATA
016144  012600                           6$:     MOV     (SP)+,R0          ;;RESTORE R0
016146  104401  001223                           TYPE    ,$CRLF            ;;''CARRIAGE RETURN'' & ''LINE FEED''
016152  000207                                   RTS     PC                ;;RETURN
016154                                   7$:
016154  013046                                   MOV     @(R0)+,-(SP)      ;;SAVE @(R0)+ FOR TYPEOUT
```

```
016156  104402                          TYPOC                          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
016160  005710                          TST     (R0)                   ;;IS THERE ANOTHER NUMBER?
016162  001770                          BEQ     6$                     ;;BR IF NO
016164  104401  016172                  TYPE    ,8$                    ;;TYPE TWO(2) SPACES
016170  000771                          BR      7$                     ;;LOOP
016172  020040     000          8$:     .ASCIZ  /  /                   ;;TWO(2) SPACES
        016176                          .EVEN
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*****************************************************************
                                .ENABL  LSB

                                .DSABL  LSB


                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                          ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE                    ;;CHARACTER IS ON THE STACK
                                ;*                                     ;;WITH PARITY BIT STRIPPED OFF
                                ;

016176  011646                  $RDCHR: MOV     (SP),-(SP)             ;;PUSH DOWN THE PC
016200  016666  000004  000002          MOV     4(SP),2(SP)            ;;SAVE THE PS
016206  105777  162732          1$:     TSTB    @$TKS                  ;;WAIT FOR
016212  100375                          BPL     1$                     ;;A CHARACTER
016214  117766  162726  000004          MOVB    @$TKB,4(SP)            ;;READ THE TTY
016222  042766  177600  000004          BIC     #^C<177>,4(SP)         ;;GET RID OF JUNK IF ANY
016230  026627  000004  000023          CMP     4(SP),#23              ;;IS IT A CONTROL-S?
016236  001013                          BNE     3$                     ;;BRANCH IF NO
016240  105777  162700          2$:     TSTB    @$TKS                  ;;WAIT FOR A CHARACTER
016244  100375                          BPL     2$                     ;;LOOP UNTIL ITS THERE
016246  117746  162674                  MOVB    @$TKB,-(SP)            ;;GET CHARACTER
016252  042716  177600                  BIC     #^C177,(SP)            ;;MAKE IT 7-BIT ASCII
016256  022627  000021                  CMP     (SP)+,#21              ;;IS IT A CONTROL-Q?
016262  001366                          BNE     2$                     ;;IF NOT DISCARD IT
016264  000750                          BR      1$                     ;;YES, RESUME
016266  026627  000004  000140  3$:     CMP     4(SP),#140             ;;IS IT UPPER CASE?
016274  002407                          BLT     4$                     ;;BRANCH IF YES
016276  026627  000004  000175          CMP     4(SP),#175             ;;IS IT A SPECIAL CHAR?
016304  003003                          BGT     4$                     ;;BRANCH IF YES
016306  042766  000040  000004          BIC     #40,4(SP)              ;;MAKE IT UPPER CASE
016314  000002                  4$:     RTI                            ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                          ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE                    ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                                     ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

016316  010346                  $RDLIN: MOV     R3,-(SP)               ;;SAVE R3
016320  012703  016424          1$:     MOV     #$TTYIN,R3             ;;GET ADDRESS
016324  022703  016442          2$:     CMP     #$TTYIN+16,R3          ;;BUFFER FULL?
016330  101405                          BLOS    4$                     ;;BR IF YES
016332  104406                          RDCHR                          ;;GO READ ONE CHARACTER FROM THE TTY
```

```
016334  112613                           MOVB    (SP)+,(R3)          ;;GET CHARACTER
016336  122713  000177           10$:    CMPB    #177,(R3)           ;;IS IT A RUBOUT
016342  001003                           BNE     3$                  ;;SKIP IF NOT
016344  104401  001222           4$:     TYPE    ,$QUES              ;;TYPE A '?'
016350  000763                           BR      1$                  ;;CLEAR THE BUFFER AND LOOP
016352  111337  016422           3$:     MOVB    (R3),9$             ;;ECHO THE CHARACTER
016356  104401  016422                   TYPE    ,9$
016362  122723  000015                   CMPB    #15,(R3)+           ;;CHECK FOR RETURN
016366  001356                           BNE     2$                  ;;LOOP IF NOT RETURN
016370  105063  177777                   CLRB    -1(R3)              ;;CLEAR RETURN (THE 15)
016374  104401  001224                   TYPE    ,$LF                ;;TYPE A LINE FEED
016400  012603                           MOV     (SP)+,R3            ;;RESTORE R3
016402  011646                           MOV     (SP),-(SP)          ;;ADJUST THE STACK AND PUT ADDRESS OF THE
016404  016666  000004  000002           MOV     4(SP),2(SP)         ;;       FIRST ASCII CHARACTER ON IT
016412  012766  016424  000004           MOV     #$TTYIN,4(SP)
016420  000002                           RTI                         ;;RETURN
016422     000                   9$:     .BYTE   0                   ;;STORAGE FOR ASCII CHAR. TO TYPE
016423     000                           .BYTE   0                   ;;TERMINATOR
016424  000016                   $TTYIN: .BLKB   16                  ;;RESERVE 16 BYTES FOR TTY INPUT
016442  052536  005015     000   $CNTLU: .ASCIZ  /^U/<15><12>        ;;CONTROL 'U'
016447     136  006507  000012   $CNTLG: .ASCIZ  /^G/<15><12>        ;;CONTROL 'G'
016454  005015  053523  020122   $MSWR:  .ASCIZ  <15><12>/SWR = /
016462  020075     000
016465     040  047040  053505   $MNEW:  .ASCIZ  /  NEW = /
016472  036440  000040
                                 .SBTTL  ROUTINE TO SIZE MEMORY

                                 ;;*****************************************************************
                                 ;*CALL:
                                 ;*      JSR     PC,$SIZE
                                 ;*      RETURN
                                 ;*$LSTAD WILL CONTAIN:
                                 ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
                                 ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
                                 ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
                                 ;*$KT11 IS THE MEMORY MANAGEMENT KEY
                                 ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
                                 ;*      MUST BE SETUP BEFORE THE CALL
                                 ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
                                 ;*      DETERMINED BY ROUTINE

016476  010046                   $SIZE:  MOV     R0,-(SP)            ;;SAVE R0 ON THE STACK
016500  010146                           MOV     R1,-(SP)            ;;SAVE R1 ON THE STACK
016502  010246                           MOV     R2,-(SP)            ;;SAVE R2 ON THE STACK
016504  010346                           MOV     R3,-(SP)            ;;SAVE R3 ON THE STACK
016506  010446                           MOV     R4,-(SP)            ;;SAVE R4 ON THE STACK             BK001
016510  013746  000114                   MOV     @#114,-(SP)         ;;SAVE MEMORY ERROR VECTOR PS & PC
016514  013746  000116                   MOV     @#116,-(SP)
016520  012737  000116  000114           MOV     #116,@#114          ;;IGNORE PARITY ERRORS WHILE SIZING
016526  012737  000002  000116           MOV     #RTI,@#116
016534  013746  000004                   MOV     @#ERRVEC,-(SP)      ;;SAVE PRESENT ERROR VECTOR PS & PC
016540  013746  000006                   MOV     @#ERRVEC+2,-(SP)
016544  010600                           MOV     SP,R0               ;;SAVE THE STACK POINTER
                                 ;;SET THE ERRVEC PS TO THE PRESENT PS
016546  104400                           TRAP                        ;;PUSH OLD PSW AND PC ON STACK
016550  012637  000006                   MOV     (SP)+,@#ERRVEC+2            ;;SAVE THE PSW IN @#ERRVEC+2
```

```
016554  012701  003776                  MOV     #3776,R1          ;;SETUP ADDRESS
016560  105727                          TSTB    (PC)+             ;;USE MEMORY MANAGEMENT?
016562  000200                 $KT11:   .WORD   200               ;;SET TO USE MEMORY MANAGEMENT
016564  100145                          BPL     $CORE             ;;BR IF NO
016566  012737  017072  000004          MOV     #$KTNEX,@#ERRVEC  ;;SET FOR TIMEOUT
016574  005737  177572                  TST     @#SR0             ;;KT11 ARE YOU THERE?
016600  052737  100000  016562          BIS     #100000,$KT11     ;;YES--SET KT11 KEY
016606  012737  016636  000004          MOV     #100$,@#ERRVEC    ;;SET FOR TIMEOUT                      BK001
016614  005737  170200                  TST     @#170200          ;;UNIBUS MAP ARE YOU THERE?            BK001
016620  012737  176200  016656          MOV     #176200,@#$STOP   ;;YES-SET COMPARISON VALUE FOR 11/70   BK001
016626  012737  000200  016654          MOV     #200,@#$MAP       ;;TURN ON MAP INDICATOR                BK001
016634  000411                          BR      $MAPRG            ;;GO SET UP MAP REGISTERS              BK001
016636  012737  006200  016656  100$:   MOV     #6200,@#$STOP     ;;COMPARISON VALUE FOR 18 BIT MAPPING  BK001
016644  022626                          CMP     (SP)+,(SP)+       ;;CLEAN OFF STACK                      BK001
016646  005037  016654                  CLR     @#$MAP            ;;MAKE SURE MAP INDICATOR TURNED OFF   BK001
016652  000412                          BR      $NOMAP            ;;                                 BK001
016654  000000                 $MAP:    .WORD   0                 ;;=200 IF MAP PRESENT                  BK001
016656  000000                 $STOP:   .WORD   0                 ;;FILLED WITH APPROPRIATE COMPARISON VALUE      BK001
016660  012703  000037         $MAPRG:  MOV     #37,R3            ;;SET UP COUNTER                       BK001
016664  012702  170200                  MOV     #170200,R2        ;;START WITH MAPL0                     BK001
016670  005022                 100$:    CLR     (R2)+             ;;LOAD ALL MAP REGISTERS               BK001
016672  012722  000074                  MOV     #74,(R2)+         ;;WITH THE VALUE 17000000              BK001
016676  077304                          SOB     R3,100$           ;;DO ALL 31 REGISTERS                  BK001
016700                         $NOMAP:
016700  005046                          CLR     -(SP)             ;;INITIALIZE FOR 'PAR'' LOADING
016702  012702  172340                  MOV     #KIPAR0,R2        ;;ADDRESS OF FIRST 'PAR''
016706  012703  000010                  MOV     #^D8,R3           ;;LOAD EIGHT 'PAR.'S'' AND EIGHT 'PDR.'S''
016712  012762  077406  177740  1$:     MOV     #77406,-40(R2)    ;;PDR = 4K, UP, READ/WRITE
016720  011622                          MOV     (SP),(R2)+        ;;LOAD 'PAR''
016722  062716  000200                  ADD     #200,(SP)         ;;UPDATE FOR NEXT 'PAR''
016726  077307                          SOB     R3,1$             ;;LOOP UNTIL ALL EIGHT ARE LOADED
016730  012742  177600                  MOV     #177600,-(R2)     ;;SETUP KIPAR7 FOR I/O
016734  005042                          CLR     -(R2)             ;;SETUP KIPAR6 FOR TESTING
016736  012737  016754  000004          MOV     #2$,@#ERRVEC      ;;CATCH TIMEOUT IF NO SR3
016744  012737  000060  172516          MOV     #60,@#SR3         ;;ENABLE 22 BIT MODE AND UNIBUS MAP    BK001
016752  000401                          BR      3$                ;;THIS PDP-11 HAS A SR3 REGISTER
016754  022626                 2$:      CMP     (SP)+,(SP)+       ;;CLEAN OFF THE STACK--NO SR3
016756  005237  177572         3$:      INC     @#SR0             ;;TURN ON MEMORY MANAGEMENT
016762  012737  017030  000004          MOV     #$KTOUT,@#ERRVEC  ;;SET FOR TIME OUT
016770  105737  016654                  TSTB    @#$MAP            ;;IS MAP THERE?                        BK001
016774  100006                          BPL     4$                ;;NO-SKIP                              BK001
016776  012737  017052  000114          MOV     #$MMOUT,@#114     ;;SET UP MEMORY ERROR VECTOR           BK001
017004  013737  000006  000116          MOV     @#ERRVEC+2,@#116  ;;LOCK OUT INTERRUPTS          BK001
017012  005737  143776         4$:      TST     @#143776          ;;TRAP ON NON-EX-MEM
017016  062712  000040                  ADD     #40,(R2)          ;;MAKE A 1K STEP
017022  023712  016656                  CMP     @#$STOP,(R2)      ;;LAST ONE?
017026  101371                          BHI     4$                ;;NO--TRY IT
017030  011202                 $KTOUT:  MOV     (R2),R2           ;;GET LAST BANK+1
017032  005037  177572                  CLR     @#SR0             ;;TURN OFF MEMORY MANAGEMENT
017036  105737  016654                  TSTB    @#$MAP            ;;IS MAP THERE?                        BK001
017042  100034                          BPL     $SIZEX            ;;NO-SKIP                              BK001
017044  005037  172516                  CLR     @#SR3             ;;TURN OFF MAP                         BK001
017050  000431                          BR      $SIZEX
017052  013704  177744         $MMOUT:  MOV     @#177744,R4       ;;SAVE MEMORY ERROR REGISTER           BK001
017056  010437  177744                  MOV     R4,@#177744       ;;CLEAR BITS IN REGISTER               BK001
017062  032704  000001                  BIT     #1,R4             ;;MEMORY TIMEOUT?                      BK001
```

```
017066  001360                             BNE     $KTOUT            ;;YES-EXIT                         BK001
017070  000002                             RTI                       ;;MUST BE PARITY ERROR-IGNORE IT   BK001
017072  042737  100000  016562  $KTNEX:  BIC     #100000,$KT11     ;;KT11 NON-EXISTENT
017100  012737  017130  000004  $CORE:   MOV     #$CROUT,@#ERRVEC  ;;SET FOR TIMEOUT
017106  005002                           CLR     R2                ;;SET UP BANK
017110  062701  004000          1$:      ADD     #4000,R1          ;;INCREMENT BY 1K
017114  062702  000040                   ADD     #40,R2            ;;1K STEP
017120  005711                           TST     (R1)              ;;TRAP ON TIME OUT
017122  022701  177776                   CMP     #177776,R1        ;;LAST ONE
017126  001370                           BNE     1$                ;;NO--TRY AGAIN
017130  162701  004000          $CROUT:  SUB     #4000,R1
017134  162702  000040          $SIZEX:  SUB     #40,R2            ;;DROP BACK
017140  010006                           MOV     R0,SP             ;;RESTORE THE STACK
017142  012637  000006                   MOV     (SP)+,@#ERRVEC+2  ;;RESTORE ERROR VECTOR
017146  012637  000004                   MOV     (SP)+,@#ERRVEC
017152  012637  000116                   MOV     (SP)+,@#116       ;;RESTORE MEMORY ERROR VECTOR
017156  012637  000114                   MOV     (SP)+,@#114
017162  010137  017206                   MOV     R1,$LSTAD         ;;LAST ADDRESS
017166  010237  017210                   MOV     R2,$LSTBK         ;;LAST BANK
017172  012604                           MOV     (SP)+,R4          ;;RESTORE R4                       BK001
017174  012603                           MOV     (SP)+,R3          ;;RESTORE R3
017176  012602                           MOV     (SP)+,R2          ;;RESTORE R2
017200  012601                           MOV     (SP)+,R1          ;;RESTORE R1
017202  012600                           MOV     (SP)+,R0          ;;RESTORE R0
017204  000207                           RTS     PC
017206  000000                  $LSTAD:  .WORD   0                 ;;CONTAINS THE LAST ADDRESS
017210  000000                  $LSTBK:  .WORD   0                 ;;CONTAINS THE LAST BANK
                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                ;;*******************************************************************
                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*        SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+14---R0

017212                          $SAVREG:
017212  010046                           MOV     R0,-(SP)          ;;PUSH R0 ON STACK
017214  010146                           MOV     R1,-(SP)          ;;PUSH R1 ON STACK
017216  010246                           MOV     R2,-(SP)          ;;PUSH R2 ON STACK
017220  010346                           MOV     R3,-(SP)          ;;PUSH R3 ON STACK
017222  010446                           MOV     R4,-(SP)          ;;PUSH R4 ON STACK
017224  010546                           MOV     R5,-(SP)          ;;PUSH R5 ON STACK
017226  016646  000022                   MOV     22(SP),-(SP)      ;;SAVE PS OF MAIN FLOW
017232  016646  000022                   MOV     22(SP),-(SP)      ;;SAVE PC OF MAIN FLOW
017236  016646  000022                   MOV     22(SP),-(SP)      ;;SAVE PS OF CALL
017242  016646  000022                   MOV     22(SP),-(SP)      ;;SAVE PC OF CALL
017246  000002                           RTI
```

```

                                          ;*RESTORE R0-R5
                                          ;*CALL:
                                          ;*        RESREG
017250                                    $RESREG:
017250  012666  000022                            MOV     (SP)+,22(SP)       ;;RESTORE PC OF CALL
017254  012666  000022                            MOV     (SP)+,22(SP)       ;;RESTORE PS OF CALL
017260  012666  000022                            MOV     (SP)+,22(SP)       ;;RESTORE PC OF MAIN FLOW
017264  012666  000022                            MOV     (SP)+,22(SP)       ;;RESTORE PS OF MAIN FLOW
017270  012605                                    MOV     (SP)+,R5           ;;POP STACK INTO R5
017272  012604                                    MOV     (SP)+,R4           ;;POP STACK INTO R4
017274  012603                                    MOV     (SP)+,R3           ;;POP STACK INTO R3
017276  012602                                    MOV     (SP)+,R2           ;;POP STACK INTO R2
017300  012601                                    MOV     (SP)+,R1           ;;POP STACK INTO R1
017302  012600                                    MOV     (SP)+,R0           ;;POP STACK INTO R0
017304  000002                                    RTI
                                          .SBTTL  TYPE ROUTINE

                                          ;;*******************************************************************
                                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                          ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                          ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                          ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                          ;*
                                          ;*CALL:
                                          ;*1) USING A TRAP INSTRUCTION
                                          ;*        TYPE     ,MESADR            ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                          ;*OR
                                          ;*        TYPE
                                          ;*        MESADR
                                          ;*

017306  105737  001157                    $TYPE:  TSTB    $TPFLG             ;;IS THERE A TERMINAL?
017312  100002                                    BPL     1$                 ;;BR IF YES
017314  000000                                    HALT                       ;;HALT HERE IF NO TERMINAL
017316  000407                                    BR      3$                 ;;LEAVE
017320  010046                            1$:     MOV     R0,-(SP)           ;;SAVE R0
017322  017600  000002                            MOV     @2(SP),R0          ;;GET ADDRESS OF ASCIZ STRING
017326  112046                            2$:     MOVB    (R0)+,-(SP)        ;;PUSH CHARACTER TO BE TYPED ONTO STACK
017330  001005                                    BNE     4$                 ;;BR IF IT ISN'T THE TERMINATOR
017332  005726                                    TST     (SP)+              ;;IF TERMINATOR POP IT OFF THE STACK
017334  012600                            60$:    MOV     (SP)+,R0           ;;RESTORE R0
017336  062716  000002                    3$:     ADD     #2,(SP)            ;;ADJUST RETURN PC
017342  000002                                    RTI                        ;;RETURN
017344  122716  000011                    4$:     CMPB    #HT,(SP)           ;;BRANCH IF <HT>
017350  001430                                    BEQ     8$
017352  122716  000200                            CMPB    #CRLF,(SP)         ;;BRANCH IF NOT <CRLF>
017356  001006                                    BNE     5$
017360  005726                                    TST     (SP)+              ;;POP  <CR><LF> EQUIV
017362  104401                                    TYPE                       ;;TYPE A CR AND LF
017364  001223                                    $CRLF
017366  105037  017574                            CLRB    $CHARCNT           ;;CLEAR CHARACTER COUNT
017372  000755                                    BR      2$                 ;;GET NEXT CHARACTER
017374  004737  017456                    5$:     JSR     PC,$TYPEC          ;;GO TYPE THIS CHARACTER
017400  123726  001156                    6$:     CMPB    $FILLC,(SP)+       ;;IS IT TIME FOR FILLER CHARS.?
```

```
017404  001350                          BNE     2$                      ;;IF NO GO GET NEXT CHAR.
017406  013746  001154                  MOV     $NULL,-(SP)             ;;GET # OF FILLER CHARS. NEEDED
                                                                        ;;AND THE NULL CHAR.
017412  105366  000001          7$:     DECB    1(SP)                   ;;DOES A NULL NEED TO BE TYPED?
017416  002770                          BLT     6$                      ;;BR IF NO--GO POP THE NULL OFF OF STACK
017420  004737  017456                  JSR     PC,$TYPEC               ;;GO TYPE A NULL
017424  105337  017574                  DECB    $CHARCNT                ;;DO NOT COUNT AS A COUNT
017430  000770                          BR      7$                      ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

017432  112716  000040          8$:     MOVB    #' ,(SP)                ;;REPLACE TAB WITH SPACE
017436  004737  017456          9$:     JSR     PC,$TYPEC               ;;TYPE A SPACE
017442  132737  000007  017574          BITB    #7,$CHARCNT             ;;BRANCH IF NOT AT
017450  001372                          BNE     9$                      ;;TAB STOP
017452  005726                          TST     (SP)+                   ;;POP SPACE OFF STACK
017454  000724                          BR      2$                      ;;GET NEXT CHARACTER
017456                          $TYPEC:
017456  105777  161462                  TSTB    @$TKS                   ;;CHAR IN KYBD BUFFER?          ;MJD001
017462  100022                          BPL     10$                     ;;BR IF NOT                     ;MJD001
017464  017746  161456                  MOV     @$TKB,-(SP)             ;;GET CHAR                      ;MJD001
017470  042716  177600                  BIC     #177600,(SP)            ;;STRIP EXTRANEOUS BITS         ;MJD001
017474  122716  000023                  CMPB    #$XOFF,(SP)             ;;WAS CHAR XOFF                 ;MJD001
017500  001012                          BNE     102$                    ;;BR IF NOT                     ;MJD001
017502                          101$:                                                                   ;MJD001
017502  105777  161436                  TSTB    @$TKS                   ;;WAIT FOR CHAR                 ;MJD001
017506  100375                          BPL     101$                                                    ;MJD001
017510  117716  161432                  MOVB    @$TKB,(SP)              ;;GET CHAR                      ;MJD001
017514  042716  177600                  BIC     #177600,(SP)            ;;STRIP IT                      ;MJD001
017520  122716  000021                  CMPB    #$XON,(SP)              ;;WAS IT XON?                   ;MJD001
017524  001366                          BNE     101$                    ;;BR IF NOT                     ;MJD001
017526                          102$:                                                                   ;MJD001
017526  005726                          TST     (SP)+                   ;;FIX STACK                     ;MJD001
017530                          10$:                                                                    ;MJD001
017530  105777  161414                  TSTB    @$TPS                   ;;WAIT UNTIL PRINTER IS READY
017534  100375                          BPL     10$                                                     ;MJD001
017536  116677  000002  161406          MOVB    2(SP),@$TPB             ;;LOAD CHAR TO BE TYPED INTO DATA REG.
017544  122766  000015  000002          CMPB    #CR,2(SP)               ;;IS CHARACTER A CARRIAGE RETURN?
017552  001003                          BNE     1$                      ;;BRANCH IF NO
017554  105037  017574                  CLRB    $CHARCNT                ;;YES--CLEAR CHARACTER COUNT
017560  000406                          BR      $TYPEX                  ;;EXIT
017562  122766  000012  000002  1$:     CMPB    #LF,2(SP)               ;;IS CHARACTER A LINE FEED?
017570  001402                          BEQ     $TYPEX                  ;;BRANCH IF YES
017572  105227                          INCB    (PC)+                   ;;COUNT THE CHARACTER
017574  000000                  $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
017576  000207                  $TYPEX: RTS     PC

                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;*******************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)               ;;NUMBER TO BE TYPED
                                ;*      TYPOS                           ;;CALL FOR TYPEOUT
```

```
                                        ;*      .BYTE   N                 ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M                 ;;M=1 OR 0
                                        ;*                                          ;;1=TYPE LEADING ZEROS
                                        ;*                                          ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)         ;;NUMBER TO BE TYPED
                                        ;*      TYPON                     ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)         ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                     ;;CALL FOR TYPEOUT

017600  017646  000000                  $TYPOS: MOV     @(SP),-(SP)       ;;PICKUP THE MODE
017604  116637  000001  020023                  MOVB    1(SP),$OFILL      ;;LOAD ZERO FILL SWITCH
017612  112637  020025                          MOVB    (SP)+,$OMODE+1    ;;NUMBER OF DIGITS TO TYPE
017616  062716  000002                          ADD     #2,(SP)           ;;ADJUST RETURN ADDRESS
017622  000406                                  BR      $TYPON
017624  112737  000001  020023          $TYPOC: MOVB    #1,$OFILL         ;;SET THE ZERO FILL SWITCH
017632  112737  000006  020025                  MOVB    #6,$OMODE+1       ;;SET FOR SIX(6) DIGITS
017640  112737  000005  020022          $TYPON: MOVB    #5,$OCNT          ;;SET THE ITERATION COUNT
017646  010346                                  MOV     R3,-(SP)          ;;SAVE R3
017650  010446                                  MOV     R4,-(SP)          ;;SAVE R4
017652  010546                                  MOV     R5,-(SP)          ;;SAVE R5
017654  113704  020025                          MOVB    $OMODE+1,R4       ;;GET THE NUMBER OF DIGITS TO TYPE
017660  005404                                  NEG     R4
017662  062704  000006                          ADD     #6,R4             ;;SUBTRACT IT FOR MAX. ALLOWED
017666  110437  020024                          MOVB    R4,$OMODE         ;;SAVE IT FOR USE
017672  113704  020023                          MOVB    $OFILL,R4         ;;GET THE ZERO FILL SWITCH
017676  016605  000012                          MOV     12(SP),R5         ;;PICKUP THE INPUT NUMBER
017702  005003                                  CLR     R3                ;;CLEAR THE OUTPUT WORD
017704  006105                          1$:     ROL     R5                ;;ROTATE MSB INTO "C"
017706  000404                                  BR      3$                ;;GO DO MSB
017710  006105                          2$:     ROL     R5                ;;FORM THIS DIGIT
017712  006105                                  ROL     R5
017714  006105                                  ROL     R5
017716  010503                                  MOV     R5,R3
017720  006103                          3$:     ROL     R3                ;;GET LSB OF THIS DIGIT
017722  105337  020024                          DECB    $OMODE            ;;TYPE THIS DIGIT?
017726  100016                                  BPL     7$                ;;BR IF NO
017730  042703  177770                          BIC     #177770,R3        ;;GET RID OF JUNK
017734  001002                                  BNE     4$                ;;TEST FOR 0
017736  005704                                  TST     R4                ;;SUPPRESS THIS 0?
017740  001403                                  BEQ     5$                ;;BR IF YES
017742  005204                          4$:     INC     R4                ;;DON'T SUPPRESS ANYMORE 0'S
017744  052703  000060                          BIS     #'0,R3            ;;MAKE THIS DIGIT ASCII
017750  052703  000040                  5$:     BIS     #' ,R3            ;;MAKE ASCII IF NOT ALREADY
017754  110337  020020                          MOVB    R3,8$             ;;SAVE FOR TYPING
017760  104401  020020                          TYPE    ,8$               ;;GO TYPE THIS DIGIT
017764  105337  020022                  7$:     DECB    $OCNT             ;;COUNT BY 1
017770  003347                                  BGT     2$                ;;BR IF MORE TO DO
017772  002402                                  BLT     6$                ;;BR IF DONE
017774  005204                                  INC     R4                ;;INSURE LAST DIGIT ISN'T A BLANK
```

```
017776  000744                          BR      2$                 ;;GO DO THE LAST DIGIT
020000  012605                  6$:     MOV     (SP)+,R5           ;;RESTORE R5
020002  012604                          MOV     (SP)+,R4           ;;RESTORE R4
020004  012603                          MOV     (SP)+,R3           ;;RESTORE R3
020006  016666  000002  000004          MOV     2(SP),4(SP)        ;;SET THE STACK FOR RETURNING
020014  012616                          MOV     (SP)+,(SP)
020016  000002                          RTI                        ;;RETURN
020020     000                  8$:     .BYTE   0                  ;;STORAGE FOR ASCII DIGIT
020021     000                          .BYTE   0                  ;;TERMINATOR FOR TYPE ROUTINE
020022     000                  $OCNT:  .BYTE   0                  ;;OCTAL DIGIT COUNTER
020023     000                  $OFILL: .BYTE   0                  ;;ZERO FILL SWITCH
020024  000000                  $OMODE: .WORD   0                  ;;NUMBER OF DIGITS TO TYPE
                                .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

020026                          $TYPDS:
020026  010046                          MOV     R0,-(SP)           ;;PUSH R0 ON STACK
020030  010146                          MOV     R1,-(SP)           ;;PUSH R1 ON STACK
020032  010246                          MOV     R2,-(SP)           ;;PUSH R2 ON STACK
020034  010346                          MOV     R3,-(SP)           ;;PUSH R3 ON STACK
020036  010546                          MOV     R5,-(SP)           ;;PUSH R5 ON STACK
020040  012746  020200                  MOV     #20200,-(SP)       ;;SET BLANK SWITCH AND SIGN
020044  016605  000020                  MOV     20(SP),R5          ;;GET THE INPUT NUMBER
020050  100004                          BPL     1$                 ;;BR IF INPUT IS POS.
020052  005405                          NEG     R5                 ;;MAKE THE BINARY NUMBER POS.
020054  112766  000055  000001          MOVB    #'-,1(SP)          ;;MAKE THE ASCII NUMBER NEG.
020062  005000                  1$:     CLR     R0                 ;;ZERO THE CONSTANTS INDEX
020064  012703  020242                  MOV     #$DBLK,R3          ;;SETUP THE OUTPUT POINTER
020070  112723  000040                  MOVB    #' ,(R3)+          ;;SET THE FIRST CHARACTER TO A BLANK
020074  005002                  2$:     CLR     R2                 ;;CLEAR THE BCD NUMBER
020076  016001  020232                  MOV     $DTBL(R0),R1       ;;GET THE CONSTANT
020102  160105                  3$:     SUB     R1,R5              ;;FORM THIS BCD DIGIT
020104  002402                          BLT     4$                 ;;BR IF DONE
020106  005202                          INC     R2                 ;;INCREASE THE BCD DIGIT BY 1
020110  000774                          BR      3$
020112  060105                  4$:     ADD     R1,R5              ;;ADD BACK THE CONSTANT
020114  005702                          TST     R2                 ;;CHECK IF BCD DIGIT=0
020116  001002                          BNE     5$                 ;;FALL THROUGH IF 0
020120  105716                          TSTB    (SP)               ;;STILL DOING LEADING 0'S?
020122  100407                          BMI     7$                 ;;BR IF YES
020124  106316                  5$:     ASLB    (SP)               ;;MSD?
020126  103003                          BCC     6$                 ;;BR IF NO
020130  116663  000001  177777          MOVB    1(SP),-1(R3)       ;;YES--SET THE SIGN
020136  052702  000060          6$:     BIS     #'0,R2             ;;MAKE THE BCD DIGIT ASCII
020142  052702  000040          7$:     BIS     #' ,R2             ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
020146  110223                          MOVB    R2,(R3)+           ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
020150  005720                          TST     (R0)+              ;;JUST INCREMENTING
```

```
020152  020027  000010                   CMP     R0,#10           ;;CHECK THE TABLE INDEX
020156  002746                           BLT     2$               ;;GO DO THE NEXT DIGIT
020160  003002                           BGT     8$               ;;GO TO EXIT
020162  010502                           MOV     R5,R2            ;;GET THE LSD
020164  000764                           BR      6$               ;;GO CHANGE TO ASCII
020166  105726                   8$:     TSTB    (SP)+            ;;WAS THE LSD THE FIRST NON-ZERO?
020170  100003                           BPL     9$               ;;BR IF NO
020172  116663  177777  177776           MOVB    -1(SP),-2(R3)    ;;YES--SET THE SIGN FOR TYPING
020200  105013                   9$:     CLRB    (R3)             ;;SET THE TERMINATOR
020202  012605                           MOV     (SP)+,R5         ;;POP STACK INTO R5
020204  012603                           MOV     (SP)+,R3         ;;POP STACK INTO R3
020206  012602                           MOV     (SP)+,R2         ;;POP STACK INTO R2
020210  012601                           MOV     (SP)+,R1         ;;POP STACK INTO R1
020212  012600                           MOV     (SP)+,R0         ;;POP STACK INTO R0
020214  104401  020242                   TYPE    ,$DBLK           ;;NOW TYPE THE NUMBER
020220  016666  000002  000004           MOV     2(SP),4(SP)      ;;ADJUST THE STACK
020226  012616                           MOV     (SP)+,(SP)
020230  000002                           RTI                      ;;RETURN TO USER
020232  023420                   $DTBL:  10000.
020234  001750                           1000.
020236  000144                           100.
020240  000012                           10.
020242  000004                   $DBLK:  .BLKW   4
                                 .SBTTL  TRAP DECODER

                                 ;;*******************************************************************
                                 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                 ;*GO TO THAT ROUTINE.

020252  010046                   $TRAP:  MOV     R0,-(SP)         ;;SAVE R0
020254  016600  000002                   MOV     2(SP),R0         ;;GET TRAP ADDRESS
020260  005740                           TST     -(R0)            ;;BACKUP BY 2
020262  111000                           MOVB    (R0),R0          ;;GET RIGHT BYTE OF TRAP
020264  006300                           ASL     R0               ;;POSITION FOR INDEXING
020266  016000  020306                   MOV     $TRPAD(R0),R0    ;;INDEX TO TABLE
020272  000200                           RTS     R0               ;;GO TO ROUTINE



                                 ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

020274  011646                   $TRAP2: MOV     (SP),-(SP)       ;;MOVE THE PC DOWN
020276  016666  000004  000002           MOV     4(SP),2(SP)      ;;MOVE THE PSW DOWN
020304  000002                           RTI                      ;;RESTORE THE PSW

                                 .SBTTL  TRAP TABLE

                                 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                 ;*BY THE "TRAP" INSTRUCTION.

                                 ;       ROUTINE
                                 ;       -------
020306  020274                   $TRPAD: .WORD   $TRAP2
020310  017306                           $TYPE   ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
020312  017624                           $TYPOC  ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
```

```
020314  017600                            $TYPOS  ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
020316  017640                            $TYPON  ;;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
020320  020026                            $TYPDS  ;;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)


020322  016176                            $RDCHR  ;;CALL=RDCHR     TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
020324  016316                            $RDLIN  ;;CALL=RDLIN     TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
020326  017212                            $SAVREG ;;CALL=SAVREG    TRAP+10(104410) SAVE R0-R5 ROUTINE
020330  017250                            $RESREG ;;CALL=RESREG    TRAP+11(104411) RESTORE R0-R5 ROUTINE
                                  .SBTTL  POWER DOWN AND UP ROUTINES

                                  ;;*****************************************************************
                                  ;POWER DOWN ROUTINE
020332  012737  020472  000024    $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
020340  012737  000340  000026            MOV     #340,@#PWRVEC+2  ;;PRIO:7
020346  010046                            MOV     R0,-(SP)         ;;PUSH R0 ON STACK
020350  010146                            MOV     R1,-(SP)         ;;PUSH R1 ON STACK
020352  010246                            MOV     R2,-(SP)         ;;PUSH R2 ON STACK
020354  010346                            MOV     R3,-(SP)         ;;PUSH R3 ON STACK
020356  010446                            MOV     R4,-(SP)         ;;PUSH R4 ON STACK
020360  010546                            MOV     R5,-(SP)         ;;PUSH R5 ON STACK
020362  017746  160552                    MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
020366  010637  020476                    MOV     SP,$SAVR6        ;;SAVE SP
020372  012737  020404  000024            MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
020400  000000                            HALT
020402  000776                            BR      .-2              ;;HANG UP

                                  ;;*****************************************************************
                                  ;POWER UP ROUTINE
020404  012737  020472  000024    $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
020412  013706  020476                    MOV     $SAVR6,SP        ;;GET SP
020416  005037  020476                    CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
020422  005237  020476            1$:     INC     $SAVR6           ;;WAIT FOR THE INC
020426  001375                            BNE     1$               ;;OF  WORD
020430  012677  160504                    MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
020434  012605                            MOV     (SP)+,R5         ;;POP STACK INTO R5
020436  012604                            MOV     (SP)+,R4         ;;POP STACK INTO R4
020440  012603                            MOV     (SP)+,R3         ;;POP STACK INTO R3
020442  012602                            MOV     (SP)+,R2         ;;POP STACK INTO R2
020444  012601                            MOV     (SP)+,R1         ;;POP STACK INTO R1
020446  012600                            MOV     (SP)+,R0         ;;POP STACK INTO R0
020450  012737  020332  000024            MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
020456  012737  000340  000026            MOV     #340,@#PWRVEC+2  ;;PRIO:7
020464  104401                            TYPE                     ;REPORT THE POWER FAILURE
020466  020500                    $PWRMG: .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
020470  000002                            RTI
020472  000000                    $ILLUP: HALT                     ;;THE POWER UP SEQUENCE WAS STARTED
020474  000776                            BR      .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
020476  000000                    $SAVR6: 0                        ;;PUT THE SP HERE
020500  005015  047520  042527    $POWER: .ASCIZ  <15><12>'POWER'
020506  000122

                                          .EVEN
                                  ;;*****************************************************************
                                  ;;*****************************************************************
                                  ;;*****************************************************************
020510                            ATEND:
 2544          000001                 .END
```

```
                                                                   J 5
UNIBUS EXERCISER        MACY11 30A(1052)  04-OCT-79  12:49  PAGE 56
CZKUAE.P11     27-SEP-79 09:25          CROSS REFERENCE TABLE -- USER SYMBOLS                                          SEQ 0061

ADD20    002532                613     622#
ADLI     010654               1083    1085     1088     1091     1757#
ALLERR   001576                467#    948
ATEND    020510                760     794      829      864      899      929      969      976     1176     1272     1281     1290     1298
                              1319    1342     1438     1442     1453     1462     1472     1498     1532     1551     1694     1775     1777
                              2543#
BE1BA    001606                473#    760*     794*     829*     864*     899*     929*     976*    1105     1176*    1240     1281*    1407
                              1442*   1694*    1711     1761*
BE1CC    001604                472#    761*     766      769      795*     800      803      830*     835      838      865*     870      873
                               900*    905      908      928*     975*    1177*    1282*    1443*    1695*    1698     1710     1760*
BE1CLR   001612                475#   1100*
BE1CR1   001610                474#    736*     737*     762*     768      796*     802      831*     837      866*     872      901*     907
                               935*    941      978*     982     1012*    1103     1180*    1283*    1310     1410     1447*    1696*    1712
                              1762*
BE1CR2   001614                476#    934*     938*     939      942      948      977*     984*     991     1084*    1086*    1087*    1089*
                              1098    1104     1132*    1135*    1148*    1178*    1179     1197*    1446*    1497*    1604     1619
BE1DB    001602                471#    514      582      641      681      985      987     1046     1709
BE1PSW   001664                496#    722*     759*     793*     828*     863*     898*     933*     974*    1011*    1081*    1129*    1280*
                              1445*
BE1VEC   001662                495#    583      721*     758*     792*     827*     862*     897*     932*     973*    1010*    1080*    1128*
                              1173*   1279*    1444*
BE2BA    001622                479#   1245     1290*    1414     1453*
BE2CC    001620                478#   1289*    1454*
BE2CLR   001626                481#
BE2CR1   001624                480#   1017*    1291*    1417     1455*
BE2CR2   001630                482#
BE2DB    001616                477#   1051
BE2PSW   001670                498#   1016*    1288*    1452*
BE2VEC   001666                497#   1015*    1287*    1451*
BE3BA    001636                485#   1250     1298*    1462*
BE3CC    001634                484#   1297*    1463*
BE3CLR   001642                487#
BE3CR1   001640                486#   1022*    1299*    1361     1422     1465*
BE3CR2   001644                488#   1464*    1531*
BE3DB    001632                483#   1056     1365     1367     1460*
BE3PSW   001674                500#   1021*    1296*    1459*
BE3VEC   001672                499#   1020*    1295*    1461*
BE4BA    001652                491#   1255     1472*
BE4CC    001650                490#   1306*    1473*
BE4CLR   001656                493#
BE4CR1   001654                492#   1028*    1308*    1381     1428     1474*
BE4CR2   001660                494#   1307*    1389*
BE4DB    001646                489#   1061     1469*
BE4PSW   001700                502#   1027*    1305*    1470*
BE4VEC   001676                501#   1026*    1304*    1471*
BGIN     002624                587     614      654#
BIT0   = 000001                122#
BIT00  = 000001                112#    122     1672     1680
BIT01  = 000002                111#    121
BIT02  = 000004                110#    120
BIT03  = 000010                109#    119      934      938
BIT04  = 000020                108#    118     1132     1135     1148     1607
BIT05  = 000040                107#    117      737     1637
BIT06  = 000100                106#    116     1163     1641
BIT07  = 000200                105#    115      564      575      660
BIT08  = 000400                104#    114     1645     1864
```

```
                                                                K  5
UNIBUS EXERCISER        MACY11 30A(1052)  04-OCT-79  12:49  PAGE 57
CZKUAE.P11     27-SEP-79 09:25            CROSS REFERENCE TABLE -- USER SYMBOLS                                     SEQ 0062

BIT09 = 001000           103#     113     1649     1874     1932
BIT1  = 000002           121#
BIT10 = 002000           102#    1653     1917
BIT11 = 004000           101#     736      991     1633     1881
BIT12 = 010000           100#    1178
BIT13 = 020000            99#    1924
BIT14 = 040000            98#     977      984     1307     1389     1446     1464     1497     1531     1850
BIT15 = 100000            97#
BIT2  = 000004           120#
BIT3  = 000010           119#
BIT4  = 000020           118#
BIT5  = 000040           117#
BIT6  = 000100           116#
BIT7  = 000200           115#
BIT8  = 000400           114#
BIT9  = 001000           113#
BKEX    010606          1732     1735#
BKGD    010534          1311     1718#
BPTVEC= 000014           129#
BRK     005364          1080     1095#
BR4TST  004212           891#
BR5TST  004050           856#
BR6TST  003706           821#
BR7TST  003544           786#
CHEX    007246          1412     1419     1424     1430#
CHKERR  010212          1626#
CLRREG  002566           638#     657      718      927      968     1007     1078     1211     1276
CLRRTN  001726           513#     578
CMPARE  005120          1034#
CNTR    010626           937     1033     1133     1749#    1763
CR    = 000015            37#    2301     2311
CRLF  = 000200            38#    2255     2311
DATA1   001714           506#    1240*    1241*    1321     1407*    1408*    1502     1506
DATA2   001716           507#    1245*    1246*    1343     1414*    1415*
DATA3   001720           508#    1250*    1251*    1533
DATA4   001722           509#    1255*    1256*    1552
DDISP = 177570            44#     263      553
DEVCNT  001702           503#     592*     615      648      668      702      705     1013     1018     1023     1034     1215     1284
                        1292     1300     1333     1352     1371     1448     1456     1466     1486     1520     1522     1542     1574
                        1610
DEVS    001704           504#    1009     1036     1037     1038     1039
DH1     011212           312      461     1783#
DH10    012017           348     1783#
DH11    012243           354     1783#
DH12    012402           359     1783#
DH13    012526           364      369     1783#
DH15    012775           375      381      386      391      396     1783#
DH2     011356           318     1783#
DH22    013545           401      406      411      416      421      426      431     1783#
DH3     011500           323      328      333      338      343     1783#
DH31    014320           437      444      449      455     1783#
DISPLA  001142           263#     553*     561*    1895*    1916*
DISPRE  000174           201#     561
DOUP    010610           971     1274     1440     1738#
DO14    005312          1082#
DSWR  = 177570            43#     262      552
```

```
                                                                L  5
UNIBUS EXERCISER        MACY11 30A(1052)  04-OCT-79  12:49  PAGE 58
CZKUAE.P11     27-SEP-79 09:25            CROSS REFERENCE TABLE -- USER SYMBOLS                                            SEQ 0063

DT1       015022         313     462    1783#
DT10      015070         350    1783#
DT11      015106         355    1783#
DT12      015122         360    1783#
DT13      015134         365     370    1783#
DT15      015142         377     382     387     392     397    1783#
DT2       015032         319    1783#
DT22      015160         402     407     412     417     422     427     432    1783#
DT3       015052         324     329     334     339     344    1783#
DT31      015172         439     445     450     456    1783#
EMTVEC= 000030           132#    536*    537*
EM1       011144         311    1783#
EM10      011751         347    1783#
EM11      012155         353    1783#
EM12      012314         358    1783#
EM13      012443         363    1783#
EM14      012547         368    1783#
EM15      012632         373    1783#
EM16      013107         380    1783#
EM17      013175         385    1783#
EM2       011243         316    1783#
EM20      013263         390    1783#
EM21      013357         395    1783#
EM22      013453         400    1783#
EM23      013607         405    1783#
EM24      013660         410    1783#
EM25      013726         415    1783#
EM26      013760         420    1783#
EM27      014022         425    1783#
EM3       011446         322    1783#
EM30      014071         430    1783#
EM31      014160         435    1783#
EM32      014413         442    1783#
EM33      014477         448    1783#
EM34      014551         453    1783#
EM35      014725         459    1783#
EM4       011561         327    1783#
EM5       011617         332    1783#
EM6       011655         337    1783#
EM7       011713         342    1783#
ENDMEM    001724         510#    516
ERRCHK    004506         721     758     792     827     862     897     932     947#    973    1128
ERRS      010474        1705    1708#
ERRTN     010166         952    1065    1107    1192    1261    1432    1597    1617#
ERRVEC= 000004           125#    550     551*    562*    579*    589*    591*    723*    931*   1855    1856*   1858*   1861*
                        2090    2091    2095*   2100*   2103*   2129*   2134*   2138    2155*   2165*   2166*
EXAD      010720        1765    1768#
EXBRK     005444        1101    1108#
EXNO      002546         616     625#
EXNOG     010472        1703    1706#
EXS       007256        1411    1418    1423    1429    1433#
FAILCK    005564        1136#
FOR4      011037         707    1783#
GNS     = ****** U       200     619     666     674     678     687     695    1809    1816    2486    2487    2488    2489
                        2490    2493    2494    2495    2496
GO        005100        1014    1019    1024    1029#
```

```
HT      = 000011          35#     2253     2311
INK       006320        1243      1248     1253     1258#
INTRTN    005250        1048      1053     1058     1063#
INTR1     005162        1010      1044#
INTR2     005200        1015      1049#
INTR3     005216        1020      1054#
INTR4     005234        1026      1059#
IOTVEC= 000020           130#      534*     535*
KIPAR0= 172340           185#     2071     2121
KIPAR1= 172342           186#
KIPAR2= 172344           187#
KIPAR3= 172346           188#
KIPAR4= 172350           189#
KIPAR5= 172352           190#
KIPAR6= 172354           191#
KIPAR7= 172356           192#
KIPDR0= 172300           174#
KIPDR1= 172302           175#
KIPDR2= 172304           176#
KIPDR3= 172306           177#
KIPDR4= 172310           178#
KIPDR5= 172312           179#
KIPDR6= 172314           180#
KIPDR7= 172316           181#
LEEV      010326        1632      1654     1656#
LF      = 000012          36#     2305     2311
LOAD1     004734        1008#
LOAD2     004772        1015#
LOAD3     005024        1020#
LOAD4     005056        1025#
LODDEV    002314         584#      608      624
MMVEC   = 000250         141#
MOVREG    002350         593#      598
MOVVEC    002414         603#
MULT1     007260        1214      1223     1232     1437#
NG        003260         719#
NODEV     002432         589       611#
NODO      006050        1165      1200#
NOG       010416         726       729      732      735     1691#
NPRTST    003402         752#
NXT       005762        1171      1182#    1199
PBERR     006014        1173      1188#
PBPSW   = 000116         470#     1175*    1183     1184*
PBRTN     006036        1174      1196#
PBVEC   = 000114         469#     1174*    1183*
PIRQ    = 177772          42#
PIRQVE= 000240           136#
PR0     = 000000          59#      775      809      844      879      914      979     1032     1278
PR1     = 000040          60#
PR2     = 000100          61#
PR3     = 000140          62#      892      936     1079     1172     1773
PR4     = 000200          63#      733      857      896      909
PR5     = 000240          64#      730      822      861      874
PR6     = 000300          65#      727      787      826      839
PR7     = 000340          66#      574      579      722      724      753      757      759      770      791      793      804      828
                         863       898      930      933      967      974     1006     1011     1016     1021     1027     1081     1127
```

```
                           1129     1166     1280     1288     1296     1305     1445     1452     1459     1470     1571
PS      = 177776            39#      40
PSW     = 177776            40#     757*     763*     791*     797*     826*     832*     861*     867*     896*     902*     930*     936*
                           967*     979*    1006*    1032*    1079*    1127*    1172*    1278*    1693*    1773*
PWRFAL    010122           655     1134     1601#
PWRVEC= 000024             131#     540*     541*     655*    1131*    1134*    2501*    2502*    2511*    2517*    2529*    2530*
QNO       010754           566     1783#
Q1        010056          1567     1585#
Q2        010066          1588#
Q3        010076          1591#
Q4        010106          1594#
RDCHR   = 104406          2039     2493#
RDLIN   = 104407          2494#
RESREG= 104411            1659     2496#
RESVEC= 000010             126#
ROTATE    010360          1222     1231     1671#
SAVREG= 104410            1618     2495#
SERV1     006202          1239#    1444
SERV2     006226          1244#    1451
SERV3     006252          1249#    1461
SERV4     006276          1254#    1471
SIMLGO    001600           468#    1031*    1476*
SR0     = 177572           145#    2101     2133*    2144*
SR1     = 177574           146#
SR2     = 177576           147#
SR3     = 172516           148#    2130*    2147*
STACK   = 001100            30#     532
START     001760           205      222      525#
STKLMT= 177774              41#
STVEC     010020           656     1565#
SWR       001140           262#     530      552*     554      560*     564      575      660     1163     1850     1864     1866     1874
                          1881     1917     1924     1929     1932     2509     2522*
SWREG     000176           202#     560
SW0     = 000001            94#
SW00    = 000001            84#      94
SW01    = 000002            83#      93
SW02    = 000004            82#      92
SW03    = 000010            81#      91
SW04    = 000020            80#      90
SW05    = 000040            79#      89
SW06    = 000100            78#      88
SW07    = 000200            77#      87
SW08    = 000400            76#      86
SW09    = 001000            75#      85
SW1     = 000002            93#
SW10    = 002000            74#
SW11    = 004000            73#
SW12    = 010000            72#
SW13    = 020000            71#
SW14    = 040000            70#
SW15    = 100000            69#
SW2     = 000004            92#
SW3     = 000010            91#
SW4     = 000020            90#
SW5     = 000040            89#
SW6     = 000100            88#
```

```
SW7    = 000200          87#
SW8    = 000400          86#
SW9    = 001000          85#
S1       007124        1279     1405#
S2       007156        1287     1413#
S3       007210        1295     1420#
S4       007226        1304     1425#
TBITVE=  000014         127#
THRU0    010342         573     1664#
TKVEC  = 000060         134#
TMPPWR   005606        1131     1146#
TPVEC  = 000064         135#
TRAPVE=  000034         133#     538*     539*
TRTVEC=  000014         128#
TSTOVR   010722        1275     1441     1772#
TST1     003252         570      717#    1832
TST10    004536         940      944      965#
TST11    004720         992      994     1005#
TST12    005262        1035     1041     1076#
TST13    005446        1092     1118#
TST14    005626        1121     1143     1162#
TST15    006052        1164     1187     1210#
TST16    006336        1216     1221     1230     1235     1271#
TST17    007074        1334     1353     1372     1393#
TST2     003400         750#
TST3     003542         756      785#
TST4     003704         790      820#
TST5     004046         825      855#
TST6     004210         860      890#
TST7     004352         895      926#
TYMOUT   002550         591      629#     723      931
TYPDS  = 104405        1813     1820     2490#
TYPE   = 104401         566      617      664      672      676      685      693      704      707     1807     1814     1821     1919
                       1927     1952     1969     1971     1974     1976     1980     1987     2043     2046     2050     2258     2371
                       2446     2486#    2531
TYPOC  = 104402        1960     1984     2487#
TYPON  = 104404        2489#
TYPOS  = 104403         669      690      698     2488#
UIPAR0=  177640         163#
UIPAR1=  177642         164#
UIPAR2=  177644         165#
UIPAR3=  177646         166#
UIPAR4=  177650         167#
UIPAR5=  177652         168#
UIPAR6=  177654         169#
UIPAR7=  177656         170#
UIPDR0=  177600         152#
UIPDR1=  177602         153#
UIPDR2=  177604         154#
UIPDR3=  177606         155#
UIPDR4=  177610         156#
UIPDR5=  177612         157#
UIPDR6=  177614         158#
UIPDR7=  177616         159#
XQ       010114        1587     1590     1593     1596#
XTST     005576        1139     1141#
```

```
$AUTOB  001134          259#
$BDADR  001122          254#
$BDDAT  001126          256#
$BELL   001216          288#    1919    1944
$CHARC  017574         2260*    2270*   2277    2303*   2308#
$CKSWR= ****** U       2493
$CMTAG  001100          242#     527     528     536     542     543     544
$CM1   = 000006         274#     275#    276#    277#    278#    279#    280#
$CM2   = 000014         274#     275#    276#    277#    278#    279#    280#
$CM3   = 000006         272#     274
$CM4   = 000006         280#     281#    282#    283#    284#    285#    286#
$CNTLG  016447         2060#
$CNTLU  016442         2059#
$CORE   017100         2099     2155#
$CRLF   001223          290#     704    1821    1927    1944    1952    1971    1976    1980    2059    2259    2311
$CROUT  017130         2155     2162#
$DBLK   020242         2412     2446    2454#
$DOAGN  015406         1803     1824    1830#
$DTBL   020232         2415     2450#
$ENDAD  015376          230     1826#   1939
$ENDCT  015242          542     1805#
$ENULL  015412         1833#
$EOP    015206         1401     1795#
$EOPCT  015234          542*    1802#   1806
$ERFLG  001103          245#     755     789     824     859     894    1220    1229    1840    1870    1872    1878*   1899
                       1914*    1944
$ERMAX  001115          251#     545*    755     789     824     859     894    1220    1229    1872    1894*   1899
$ERROR  015674          536     1913#
$ERRPC  001116          252#    1783    1921*   1922*   1923    1944    1958
$ERRTB  001226          307#    1966
$ERRTY  016042         1926     1951#
$ERTTL  001112          249#    1818    1822*   1920*   1944
$ESCAP  001214          287#     544*   1893*   1935    1937    1944
$FILLC  001156          270#    2263    2311
$FILLS  001155          269#    2311
$GDADR  001120          253#
$GDDAT  001124          255#
$GET42  015366         1823#
$GTSWR= ****** U       2492
$HD    = 000001          14       15
$ICNT   001104          246#    1783    1885*   1886    1888*   1898
$ILLUP  020472         2501     2517    2534#
$INTAG  001135          260#
$ITEMB  001114          250#    1923*   1944    1955
$KTNEX  017072         2100     2154#
$KTOUT  017030         2134     2143#   2152
$KT11   016562         2098#    2102*   2154*
$LF     001224          291#    1944    2050    2059    2311
$LPADR  001106          247#     546*   1876*   1891*   1896    1898
$LPERR  001110          248#     547*   1876    1892*   1898    1934
$LSTAD  017206         2169*    2177#
$LSTBK  017210         2170*    2178#
$MAIL = ****** U        564     1891    1929    2247
$MAP    016654         2106*    2110*   2112#   2135    2145
$MAPRG  016660         2107     2114#
$MMOUT  017052         2137     2149#
```

```
                                                                            D  6
UNIBUS EXERCISER        MACY11 30A(1052)  04-OCT-79  12:49  PAGE 63
CZKUAE.P11    27-SEP-79 09:25          CROSS REFERENCE TABLE -- USER SYMBOLS                                SEQ 0068

$MNEW   016465          2063#
$MSWR   016454          2061#
$MXCNT  015672          1889    1898#
$NOMAP  016700          2111    2119#
$NULL   001154           268#   1137    1147    2265    2311
$NWTST= 000001           711#    713     741#    743     778#    780     813#    815     848#    850     883#    885     918#
                         920     957#    959     997#    999    1069#   1071    1111#   1113    1152#   1154    1203#   1205
                        1265#   1267    1390#
$OCNT   020022          2343*   2372*   2385#
$OMODE  020024          2338*   2342*   2347    2350*   2361*   2387#
$OVER   015656          1851    1869    1877    1887    1895#
$PASS   001100           243#   1799*   1800*   1811    1833    1883    1899
$POWER  020500          1142    2532    2537#
$PWRDN  020332           540    1149    1614    2501#   2529
$PWRMG  020466          1137    1142*   1147*   2532#
$PWRUP  020404          2511    2517#
$QUES   001222           289#   1944    2043    2059    2311
$RDCHR  016176          2007#   2493
$RDDEC= ****** U        2495
$RDLIN  016316          2035#   2494
$RDOCT= ****** U        2495
$RDSZ = 000016          2028#
$REGAD  001160           272#
$REG0   001162           274#    518     659*    684*    697     768*    802*    837*    872*    907*    941*   1329*   1348*
                        1516*   1539*   1558*   1709*   1783
$REG1   001164           275#    769*    803*    838*    873*    908*    942*    987*   1036*   1103*   1328*   1347*   1367*
                        1385*   1482*   1483    1515*   1538*   1557*   1710*   1719*   1720    1783
$REG2   001166           276#    631*    632*    770*    804*    839*    874*    909*    988*   1037*   1104*   1259*   1320*
                        1323    1325*   1330    1368*   1386*   1486    1665*   1666*   1711*   1733*   1774*   1783
$REG3   001170           277#    771*    805*    840*    875*    910*   1038*   1105*   1330*   1349*   1517*   1537*   1556*
                        1712*   1783
$REG4   001172           278#   1039*   1303*   1383    1385    1426*   1713*   1750*   1751*   1752    1783
$REG5   001174           279#    594*    950*   1064*   1096*   1189*   1260*   1431*   1783
$RESRE  017250          2212#   2496
$RTNAD  015410          1832#
$R2A  = ****** U        2497
$SAVRE  017212          2196#   2495
$SAVR6  020476          2510*   2518    2519*   2520*   2536#
$SCOPE  015416           534    1849#
$SETUP= 000037           194#    533     534     536     538     540     542     543     544     546    1797    1850    1914
                        1932    1939    1996    2065
$SIZE   016476          2081#
$SIZEX  017134          2146    2148    2163#
$STOP   016656          2105*   2108*   2113#   2141
$STUP = 177777           194#
$SVLAD  015630          1859    1890#
$SVPC = 000232           228#    233
$SWR  = 167400             2#     14      19      20      21      22      23      24      25     286     287     288     543
                         544     546     547     718     751     786     821     856     891     927     966    1006    1077
                        1119    1163    1211    1272    1394    1792    1798    1825    1831    1833    1841    1842    1843
                        1844    1845    1850    1862    1864    1865    1870    1871    1872    1879    1880    1881    1892
                        1895    1898    1905    1906    1907    1908    1909    1917    1924    1929    1932    1944    2533
$SWRMK= 000300             3#     25      26    1845    1846    1866    1867
$TIMES  001212           286#    543*   1119*   1394*   1798*   1879*   1886    1889*   1898
$TKB    001146           265#   1396    1994    2011    2017    2284    2291
$TKS    001144           264#   1994    2009    2015    2282    2289
```

```
$TMP0    001176             204*     221*     280#     568      596*     597      599*     658*     680*     683*     689      702
$TMP1    001200             281#    1130*
$TMP2    001202             282#
$TMP3    001204             283#    1167*    1169*    1170
$TMP4    001206             284#     951*    1047*    1052*    1057*    1062*    1097*    1126*    1191*    1242*    1247*    1252*    1257*
                           1409*    1416*    1421*    1427*    1586*    1589*    1592*    1595*    1624     1783
$TMP5    001210             285#     520
$TN    = 000020              14#     711      718#     741      751#     756      778      786#     790      813      821#     825      848
                            856#     860      883      891#     895      918      927#     940      944      957      966#     992      994
                            997     1006#    1035     1041     1069     1077#    1092     1111     1119#    1121     1143     1152     1163#
                           1187     1203     1211#    1216     1221     1230     1235     1265     1272#    1334     1353     1372     1390
                           1394#
$TPB     001152             267#    2300*    2311
$TPFLG   001157             271#    2241     2311
$TPS     001150             266#    2298     2311
$TRAP    020252             538     2463#
$TRAP2   020274            2474#    2485
$TRP   = 000012            2478#    2487#    2488#    2489#    2490#    2491#    2493     2494#    2495#    2496#    2497#
$TRPAD   020306            2468     2485#
$TSTNM   001102             244#    1783     1797*    1840     1868     1890*    1895     1899     1916     1944
$TTYIN   016424            2036     2037     2054     2058#
$TYPBN=  ****** U          2491
$TYPDS   020026            2400#    2490
$TYPE    017306            2241#    2478     2486
$TYPEC   017456            2262     2269     2276     2281#
$TYPEX   017576            2304     2306     2309#
$TYPOC   017624            2341#    2487
$TYPON   017640            2340     2343#    2489
$TYPOS   017600            2336#    2488
$XOFF  = 000023            2286     2311
$XON   = 000021            2293     2311
$XTSTR   015426            1853#
$$GET4=  000000            1825#
$OFILL   020023            2337*    2341*    2351     2386#
$40CAT=  ****** U          1850     1926
.      = 020510             196#     200#     203#     220#     228      229#     231#     233#     241#     292      531      546      547
                            620#     675#     679#     688#    1783#    1810#    1833     1834#    1898     1899     1944     1990#    1994
                           2058#    2059     2065     2311     2454#    2513     2535
```

```
                                                                              F  6
UNIBUS EXERCISER        MACY11 30A(1052)  04-OCT-79  12:49  PAGE 66
CZKUAE.P11     27-SEP-79 09:25           CROSS REFERENCE TABLE -- MACRO NAMES                                SEQ 0070

BE1       751#    754     788     823     858     893
COMMEN      1#    137#    207    1312    1335    1354    1373    1477    1489    1524    1544
ENDCOM      1#    137#    216    1316    1339    1358    1377    1481    1494    1528    1548
ERROR      31#    633     772     806     841     876     911     943     989     993    1040    1106    1140    1190    1331
         1350    1369    1387    1518    1540    1559    1629    1635    1639    1643    1647    1651    1655    1667    1714
         1734
ESCAPE      1#    137#
GETPRI      1#    137#   2094
GETSWR      1#    137#
MSSG      710#    713     741#    743     777#    780     812#    815     847#    850     882#    885     917#    920     956#
          959     996#    999    1068#   1071    1110#   1113    1151#   1154    1202#   1205    1265#   1267
MULT        1#    137#
NEWTST      1#    137#    711     741     778     813     848     883     918     957     997    1069    1111    1152    1203
         1265    1390
NOGRNT    740#   1692
POP         1#    137#   2217    2441    2522    2523
PUSH        1#    137#   2197    2400    2503    2509
REPORT      1#    137#
SCOPE      32#    717     750     785     820     855     890     926     965    1005    1076    1118    1162    1210    1271
         1393    1796
SETPRI      1#    137#
SETTRA   2478#   2487    2488    2489    2490    2493    2494    2495    2496
SETUP       1#    137#    526
SKIP        1#    137#    756     790     825     860     895     940     944     992     994    1035    1041    1092    1121
         1143    1187    1216    1221    1230    1235    1334    1353    1372
SLASH       1#    137#    651    1562
SPACE     137#
STARS       1#    137#    226     237     292     308     464     511     523     609     627     635     711     716     741
          749     778     784     813     819     848     854     883     889     916     918     925     945     957     964
          997    1004    1042    1069    1075    1093    1111    1117    1144    1152    1161    1194    1203    1209    1236
         1263    1265    1270    1390    1392    1402    1404    1435    1488    1577    1599    1615    1661    1669    1689
         1707    1716    1717    1736    1746    1755    1770    1780    1783    1788    1837    1901    1946    1993    1999
         2028    2067    2181    2226    2313    2390    2457    2499    2515    2540
SWRSU       1#    137#    548#
TRMTRP   2478#
TYPBIN      1#    137#
TYPDEC      1#    137#   1811    1818
TYPNAM      1#    137#
TYPNUM      1#    137#
TYPOCS      1#    137#    668     689     697
TYPOCT      1#    137#   1958    1982
TYPTXT      1#    137#    617     664     672     676     685     693    1807    1814
$$CMRE    235#    274     275     276     277     278     279
$$CMTM    235#    280     281     282     283     284     285
$$ESCA      1#    137#
$$NEWT      1#    137#    711     741     778     813     848     883     918     957     997    1069    1111    1152    1203
         1265    1390
$$SET    2478#   2487    2488    2489    2490    2493    2494    2495    2496
$$SKIP      1#    137#    756     790     825     860     895     940     944     992     994    1035    1041    1092    1121
         1143    1187    1216    1221    1230    1235    1334    1353    1372
.EQUAT      1#      2#     27
.HEADE      1#      2#      4
.KT11       1#      2#    137
.SETUP      1#      2#    194
.SWRHI      1#      2#     15
.SWRLO      2#     26#
```

```
.$ACT1       1#        2#     224
.$APTB       1#
.$APTH       1#
.$APTY       1#
.$ASTA       1#
.$CATC       1#        2#     194
.$CMTA       1#        2#     235
.$DB2D       1#
.$DB2O       1#
.$DIV        1#
.$EOP        1#        2#    1786
.$ERRO       1#        2#    1899
.$ERRT       1#        2#    1944
.$MULT       1#
.$POWE       1#        2#    2497
.$RAND       1#
.$RDDE       1#
.$RDOC       1#
.$READ       1#        2#    1991
.$R2AZ       1#
.$SAVE       1#        2#    2179
.$SB2D       1#
.$SB2O       1#
.$SCOP       1#        2#    1835
.$SIZE       1#        2#    2065
.$SUPR       1#
.$TRAP       1#        2#    2455
.$TYPB       1#
.$TYPD       1#        2#    2388
.$TYPE       1#        2#    2224
.$TYPO       1#        2#    2311
.$40CA       1#
.1170        1#

. ABS.  020510      000

 ERRORS DETECTED:  0

 CZKUAE,CZKUAE.LST/CRF/SOL=[400,4531]SYSMAC.C4,[400,2465]CZKUAE.P11
 RUN-TIME: 42 51 3 SECONDS
 RUN-TIME RATIO: 472/98=4.8
 CORE USED:  33K  (65 PAGES)
```