IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DXQLA-I-D

PRODUCT NAME: DEC/X11 MONITOR LIBRARY

DATE: MARCH 1977

MAINTAINER: DIAGNOSTIC GROUP

CONTENTS

```
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;               IDENTIFICATION
                        ;
                        ;PRODUCT CODE:  MAINDEC-11-DXQAB-M-LA
                        ;
                        ;PRODUCT NAME:  DECX11 - STANDARD MONITOR
                        ;
                        ;DATE:          APRIL 1977
                        ;
                        ;MAINTAINER:    DIAGNOSTIC GROUP
                        ;
                        ;
                        ;
                        ;COPYRIGHT 1973, 1974, 1975, 1976, 1977  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```



```
                                .LIST   SEQ
                                .TITLE  QABM  DEC/X11 STANDARD MONITOR
000000                          .ASECT
000000                          .=0

                                .GLOBL  LCOR12,HCOR12,OACNV,BDCNV,MODQ,CLOCKL,CLOCKP
                                .GLOBL  KW11L,KW11P,HMS,HOURS,WCASE,WCASEE,CLOCK
                                .GLOBL  OFFSET,HUNGCK,HNGTIM,LCLEAR,PCLEAR,CHKSUM
                                .GLOBL  TKS,TKB,TPS,TPB,FILCNT,FILLER,SP,RANNUM,CAST
                                .GLOBL  MODCNT,MODCTR,PASREL,RELOC,ALLBK,KTPRES,PWRCNT

                        ;******IF RUNNING WITH AN 11/60 AND ANY SYSTEM TYPE ERROR OCCURS
                        ;       (TRAP TO 4, ETC.) "LOGBUF" WILL CONTAIN
                        ;       THE CONTENTS OF SCRATCHPAD LOCATIONS FROM 0-177, SCRATCHPAD
                        ;       LOCATIONS 100-107 ARE PRINTED OUT.

                        ;SWITCH REGISTER OPTIONS:

                                ;SR15=1  HALT MODULE AFTER ERROR.
                                ;SR14=1  INHIBIT MODULE HALT AFTER 20 ERRORS.
                                ;SR13=1  INHIBIT ERROR PRINT.
                                ;SR12=1  ENABLE ENDPAS PRINTOUT.
                                ;SR10=1  PRINT ALL DATA ERRORS IN BUFFER
                                ;SR9=1   INHIBIT "RELOCATED TO " MESSAGE

                        ;MODULE STATUS BIT DEFINITIONS:

                        ;       BIT 15=1        IOMOD
                        ;       BIT 14=1        MODULE SELECTED
                        ;       BIT 13=1        MODULE DROPPED
                        ;       BIT 12=1        EXTENDED MODULE
                        ;       BIT 11=1        DESELECT IF PROGRAM IS RELOCATED
                        ;       BIT 10=1        DESELECT IF NOT ON AN EVEN 32K OFFSET
                        ;       BIT 09=1        NBKMOD
                        ;       ALL 0'S         SBKMOD
```

```
                        ;A FEW DEFINITIONS.
000000                  R0      =%0     ;GENERAL PURPOSE REGISTER 0
000001                  R1      =%1     ;GENERAL PURPOSE REGISTER 1
000002                  R2      =%2     ;GENERAL PURPOSE REGISTER 2
000003                  R3      =%3     ;GENERAL PURPOSE REGISTER 3
000004                  R4      =%4     ;GENERAL PURPOSE REGISTER 4
000005                  R5      =%5     ;GENERAL PURPOSE REGISTER 5
000006                  R6      =%6     ;GENERAL PURPOSE REGISTER 6
000006                  SP      =%6     ;POINTER TO HARDWARE (SYSTEM DEFAULT) STACK
000007                  PC      =%7     ;PROGRAM COUNTER

177776                  PS      =177776 ;PROCESSOR STATUS REGISTER
177776                  PSW     =177776 ;PROCESSOR STATUS WORD
177570                  HARDSR  =177570 ;HARDWARE SWITCH REGISTER

177572                  SSR0    =177572 ;MEMORY MANAGEMENT REGISTER 0
172516                  SSR3    =172516 ;MEMORY MANAGEMENT REGISTER 3

170200                  MAPREG  =170200 ;BASE ADDRESS OF UNIBUS MAP BOX (11/70)
```



```
076600                  MED     =76600  ;MAINTENANCE EXAMINE AND DEPOSIT INSTRUCTION (11/60)
000101                  RDSERV  =101    ;LOG SERVICE REG. ADR -- READ
000102                  RDPBA   =102    ;LOG PHYSICAL BUSS ADR. REG. ADR. -- READ
000103                  RDCUA   =103    ;LOG CURRENT MICRO ADR. REG. ADR. -- READ
000104                  RDFGIN  =104    ;LOG FLAG/INTERRUPT REG. ADR. -- READ
000022                  RDWHMI  =022    ;GEN. WHO AM I REG. ADR. -- READ
000222                  WTWHMI  =222    ;GEN. WHO AM I REG. ADR. -- WRITE

177772                  PIRQRG  =177772 ;PROGRAMMED INTERRUPT REQUEST REGISTER

140000                  UMODE   =140000 ;BIT CONFIGURATION TO INDICATE USER MODE IN PSW
040000                  SMODE   =40000  ;BIT CONFIGURATION TO INDICATE SUPERVISOR MODE IN PSW

172300                  KIPDR0  =172300 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 0
172302                  KIPDR1  =172302 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 1
172304                  KIPDR2  =172304 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 2
172306                  KIPDR3  =172306 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 3
172316                  KIPDR7  =172316 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 7
172340                  KIPAR0  =172340 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 0
172342                  KIPAR1  =172342 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 1
172344                  KIPAR2  =172344 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 2
172346                  KIPAR3  =172346 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 3
172356                  KIPAR7  =172356 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 7

177600                  UIPDR0  =177600 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 0
177602                  UIPDR1  =177602 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 1
177604                  UIPDR2  =177604 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 2
177616                  UIPDR7  =177616 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 7
177640                  UIPAR0  =177640 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 0
177642                  UIPAR1  =177642 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 1
177644                  UIPAR2  =177644 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 2
177656                  UIPAR7  =177656 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 7
```

```
                                PIRQ$   =IOT     ;MAKE THE PIRQ CALL A TRAP THROUGH THE IOT VECTOR
                                OPEN    =0      ;USE OPEN TO DECLARE ALL VARIABLES AS 0 BEFORE STARTING
                                BIT15   =100000 ;BIT 15 DEFINITION
                                BIT14   =40000  ;BIT 14 DEFINITION
                                BIT13   =20000  ;BIT 13 DEFINITION
                                BIT12   =10000  ;BIT 12 DEFINITION
                                BIT11   =4000   ;BIT 11 DEFINITION
                                BIT10   =2000   ;BIT 10 DEFINITION
                                BIT9    =1000   ;BIT  9 DEFINITION
                                BIT8    =400    ;BIT  8 DEFINITION
                                BIT7    =200    ;BIT  7 DEFINITION
                                BIT6    =100    ;BIT  6 DEFINITION
                                BIT5    =40     ;BIT  5 DEFINITION
                                BIT4    =20     ;BIT  4 DEFINITION
                                BIT3    =10     ;BIT  3 DEFINITION
                                BIT2    =4      ;BIT  2 DEFINITION
                                BIT1    =2      ;BIT  1 DEFINITION
                                BIT0    =1      ;BIT  0 DEFINITION

                                PRTY7   =340    ;BIT CONFIGURATION FOR PRIORITY LEVEL 7 IN PSW
                                PRTY6   =300    ;BIT CONFIGURATION FOR PRIORITY LEVEL 6 IN PSW
                                PRTY5   =240    ;BIT CONFIGURATION FOR PRIORITY LEVEL 5 IN PSW
                                PRTY4   =200    ;BIT CONFIGURATION FOR PRIORITY LEVEL 4 IN PSW

                                PUSH    =005746 ;TST    -(SP)   ;;USED TO ADD A WORD TO  THE STACK
                                PUSH2   =024646 ;CMP    -(SP),-(SP)     ;;USED TO PUT TWO WORDS ON THE STACK
                                POPSP   =005726 ;TST    (SP)+   ;;USED TO DELETE A WORD FROM THE STACK
                                POPSP2  =022626 ;CMP    (SP)+,(SP)+;USED TO DELETE TWO WORDS FROM THE STACK

                                IE      =BIT6   ;DEFINITION OF INTERRUPT ENABLE BIT FOR ALL PDP-11 DEVICES
                                KBUFL   =32.    ;DEFINITION OF KEYBOARD BUFFER LENGTH (IN WORDS)
                                IOQL    =400.   ;DEFINITION OF INPUT/OUTPUT SERVICE QUEUE LENGTH(IN BYTES)
                                TYPQL   =600.   ;DEFINITION OF TYPING QUEUE LENGTH (IN BYTES)
                                TYPLIM  =TYPEQ+TYPQL    ;DEFINITION OF LAST WORD IN TYPE QUEUE
                                IOQLIM  =IOQ+IOQL       ;DEFINITION OF LAST WORD IN I/O QUEUE
```



```
                                ;THE FOLLOWING DEFINITIONS ARE THE INDEX VALUES USED BY MODE 6 AND 7
                                ;MONITOR INSTRUCTIONS WHEN REFERRING TO LOCATIONS IN THE HEADER OF A MODULE

000005                          XFLAG   =5.     ;REFERENCE USED TO KEEP TRACK OF WRITE BUFFER USAGE
000020                          STAT    =16.    ;POINTS TO THE LOW BYTE OF THE MODULE STATUS WORD
000021                          STAT1   =17.    ;POINTS TO THE HIGH BYTE OF THE MODULE STATUS WORD
000022                          INIT    =18.    ;POINTS TO THE MODULE 'S STARTING ADDRESS
000024                          SPOINT  =20.    ;POINTS TO THE MODULE'S STACK POINTER
000026                          PSCNT   =22.    ;REFERENCES THE LOCATION KEEPING TRACK OF THE NUMBER OF PASSES
000030                          EPCNT   =24.    ;REFERENCES THE LOCATION KEEPING TRACK OF THE NUMBER OF ERRORS
000032                          SVR0    =26.    ;POINTS TO WHERE THE MODULE'S REGISTER 0 CONTENTS ARE STORED
000034                          SVR1    =28.    ;POINTS TO WHERE THE MODULE'S REGISTER 1 CONTENTS ARE STORED
000036                          SVR2    =30.    ;POINTS TO WHERE THE MODULE'S REGISTER 2 CONTENTS ARE STORED
000040                          SVR3    =32.    ;POINTS TO WHERE THE MODULE'S REGISTER 3 CONTENTS ARE STORED
000042                          SVR4    =34.    ;POINTS TO WHERE THE MODULE'S REGISTER 4 CONTENTS ARE STORED
000044                          SVR5    =36.    ;POINTS TO WHERE THE MODULE'S REGISTER 5 CONTENTS ARE STORED
000046                          SVR6    =38.    ;POINTS TO WHERE THE MODULE'S REGISTER 6 CONTENTS ARE STORED
000050                          CSRA    =40.    ;REFERENCES THE BASE ADDRESS OF THE MODULE'S DEVICES(CONTROL AND
000052                          ACSR    =42.    ;REFERENCES THE CONTENTS OF THE CONTROL AND STATUS REGISTER
000052                          SBADR   =42.    ;POINTS TO THA ADDRESS OF WHAT THE DATA SHOULD HAVE BEEN
000054                          ASTAT   =44.    ;POINTS TO ANYSTATUS REGISTERNOT THE SAME AS THE CSR
000054                          WASADR  =44.    ;POINTS TO THE ADDRESS OF WHAT THE DATAWAS
000056                          ASB     =46.    ;POINTS TO THEACTUAL EXPECTED DATA
000060                          AWAS    =48.    ;POINTS TO THE ACTUAL RECEIVED DATA
000062                          RSTRT   =50.    ;POINTS TO THE MODULE RESTART ADDRESS
000064                          RBUFVA  =52.    ;REFERENCES THE VIRTUAL ADDRESS OF THE READ BUFFER
000066                          RBUFPA  =54.    ;POINTS THE LOCATION FOR THE PHYSICAL ADDRESS OF THE READ BUFFER
000070                          RBUFEA  =56.    ;POINTS THE EXTENDED ADDRESS BITS OF THE READ BUFFER
000072                          RBUFSZ  =58.    ;POINTS THE SIZE OF THE READ BUFFER
000074                          WBUFPA  =60.    ;POINTS TO THE WRITE BUFFER PHYSICAL ADDRESS
000076                          WBUFEA  =62.    ;POINTS TO THE EXTENDED ADDRESS BITS OF THE WRITE BUFFER
000100                          WBUFRQ  =64.    ;REFERENCES THE AMOUNT OF WRITE BUFFER REQUESTED BY MODULE
000102                          WBUFSZ  =66.    ;REFERENCES THE SIZE OF THE WRITE BUFFER
```

```
                                ;LOAD TRAP AND VECTOR AREA

                        000000          .=0
000000  000000  000000          .WORD   0,0
000004  015246          BUSEV:  BUSERR                  ;BUS ERROR TRAP POINTER.
000006  000340                  PRTY7
000010  015276          RESIV:  RESINT                  ;RESERVED INSTRUCTION TRAP.
000012  000340                  PRTY7
000014  005700          TRCV:   TRCI                    ;TRACE TRAP POINTER.
000016  000000                  0
000020  006416          IOTV:   PIRQ.                   ;SERVICE PIRQ CALL
000022  000340                  PRTY7                   ;PREVENT LOWER LEVEL MODULE FROM COMING IN ON
                                                        ;TOP OF HIGHER LEVEL ONES
000024  015460          PWRFV:  PWRDN                   ;POWER FAIL POINTER.
000026  000340                  PRTY7
000030  000032          EMTV:   .+2                     ;EMT POINTER
000032  000000                  HALT
000034  015134          TRPV:   TRPINT                  ;TRAP POINTER.
000036  000000                  0
                        000040          .=40
000040  000000                  .WORD   OPEN            ;LOAD MEDIUM INDICATOR.
000042  000000          DDPPTR: .WORD   0               ;CHAIN MODE ONLY. POINTS TO DDPMON.
000044  000000                  .WORD   0
000046  016134                  LOGIC
000050  000000                  OPEN
000052  000000                  OPEN
000054  000000G                 CAST                    ;GLOBAL POINTER FOR CAST MODIFICATION
000056  000001                  1                       ;ALSO FOR CAST
                        000060          .=60
000060  006552          KYBDV:  KBSRVC                  ;KYBD INTERRUPT POINTER.
000062  000340                  PRTY7
000064  001526          TTPIV:  TTYSRV                  ;TELETYPE PRINTER SERVICE
000066  000200                  PRTY4
```



```
                                ;FROM 70 THROUGH 776 FILLED WITH .+2 AND HALT.


                        000114          .=114
000114  015672          PARVCT: PARERR          ;PARITY ERROR INTERUPT POINTER
000116  000340                  PRTY7


                                ;LOAD STARTING ADDRESSES

                        000200          .=200
000200  000167  001330          JMP     START           ;GO TO START OF MONITOR.
                        001000          .=1000
001000  000167  000530          JMP     START           ;GO TO START OF MONITOR.


001004  000000          FAKESR: OPEN                    ;LOCATION OF PSEUDO SWITCH REG, WHEN USED
```

```
                                ;VARIABLE DATA AREA CLEARED EVERY TIME RUN IS ISSUED OR ^C IS TYPED

001006  000                    IOQUE:  .BYTE   OPEN            ;IOQUE AND TYPQUE MUST BE IN SAME WORD!!
001007  000                    TYPQUE: .BYTE   OPEN
001010  000                    SPCFLG: .BYTE   OPEN            ;SPCFLG AND DIRIND MUST BE IN SAME WORD!!
001011  000                    DIRIND: .BYTE   OPEN
001012  000                    BKQUE:  .BYTE   OPEN
001013  000                    BRAKE:  .BYTE   OPEN
001014  000                    TTYBSY: .BYTE   OPEN            ;TELETYPE BUSY FLAG, 0= NOT BUSY,
001015  000                    MODCNT: .BYTE   OPEN
001016  000                    MODCTR: .BYTE   OPEN
001017  000                    ERRIND: .BYTE   OPEN            ;0=ERROR, NOT 0=DATA ERROR,
001020  000                    FILCTR: .BYTE   OPEN
001021  000                    MDXCTR: .BYTE   OPEN            ;EXTENDED I/O MODULES NOT YET RUN IN CURRENT BANK
001022  000                    MDXCNT: .BYTE   OPEN            ;EXTENDED I/O MODULE COUNT
001023  000                    WBFLG:  .BYTE   OPEN            ;CURRENT VALUE OF XFLAG BEING CHECKED FOR
001024  000                    STOP:   .BYTE   OPEN            ;STOP MODULES TO RELOCATE PROGRAM
001025  000                    BKCNT:  .BYTE   OPEN            ;BACKGROUND MODULE COUNT
001026  000                    XCNT:   .BYTE   OPEN            ;RUNNABLE EXTENDED MODULE COUNT (TO CALCULATE ROTCT)
001027  000                    QUECTR: .BYTE   OPEN
001030  000                    XFRFLG: .BYTE   OPEN
001031  000                    ROTCT:  .BYTE   OPEN            ;VALUE USED FOR BUFFER ROTATION TIMEOUT (16 TIMES
                                                               ;XCNT OR 377, WHICHEVER IS SMALLER)
001032  000                    PASTOT: .BYTE   OPEN            ;ALTERNATE RELOCATION COUNTER= TOTAL ENDPASS
                                                               ;CALLS IN CURRENT BANK
001033  000                    PASREL: .BYTE   OPEN
001034  000                    BKTIMR: .BYTE   OPEN            ;BKMOD INSTR CTR= LOADED FROM BKTIME
001035  000                    ROT:    .BYTE   OPEN            ;WBUF ROTATION RATE CTR= LOADED FROM ROTN
                               .EVEN
001036  000000                 KBPTR:  OPEN
001040  000000                 MODPTR: OPEN                    ;MODULE POINTER
001042  000000                 ADDRLO: OPEN
001044  000000                 ADDRHI: OPEN
001046  000000                 NUMBER: OPEN
001050  000000                 NUMBPA: OPEN
001052  000000                 NUMBEA: OPEN
001054  000000                 DSTADR: OPEN
001056  000000                 SRETRN: OPEN
001060  000000                 IOBKID: OPEN
001062  000000                 YES:    OPEN
001064  000000                 TABADR: OPEN
001066  000000                 RSTAT:  OPEN
001070  000000                 TRCPC:  OPEN
001072  000000                 TRCPSW: OPEN
001074  000000                 IOQ1:   OPEN                    ;I/O QUEUE POINTERS,
001076  000000                 IOQ2:   OPEN
001100  000000                 TYPQ1:  OPEN                    ;LOAD TYPE QUEUE POINTER
001102  000000                 TYPQ2:  OPEN                    ;UNLOAD TYPE QUEUE POINTER
001104  000000                 TYPADR: OPEN
001106  000000                 TYPRET: OPEN
001110  000000                 TYPCTR: OPEN                    ;USED TO MONITOR CONSOLE TELETYPE TO PREVENT HANGING
001112  000000                 OASCEA: OPEN                    ;USED TO HOLD EA BITS FOR OCTAL TO
                                                               ;ASCII CONVERSION

001114  000000                 MODWPA: OPEN
001116  000000                 MODWEA: OPEN
```



```
001120  000000                 MODW12: OPEN
001122  000000                 MODRPA: OPEN
001124  000000                 MODREA: OPEN
001126  000000                 MODR12: OPEN
001130  000000                 DERRCT: OPEN
001132  000000                 SAVLO:  OPEN
001134  000000                 SAVHI:  OPEN
001136  000000                 MODADR: OPEN                    ;ADDRESS OF CURRENTLY EXECUTING (IOQ LEVEL) MODULE
001140  000000                 TYPP1:  OPEN                    ;SAVE LOCATIONS FOR TYPE ROUTINE REGISTERS
001142  000000                 TYPR2:  OPEN
001144  000000                 TYPR3:  OPEN
001146  000000                 TYPR4:  OPEN
001150  000000                 TYPR5:  OPEN
001152  000000                 SPSAV:  OPEN
001154  000000                 ERRPOS: OPEN
001156  000000                 CCFLG:  OPEN                    ;INDICATE RETURN TO CHECKDATA ROUTINE
001160  000000                 CCSRA:  OPEN
```

```
                                ;VARIABLE AND FIXED DATA AREA NOT CLEARED WHEN RUN OR ^C IS ISSUED

001162  177560                  TKS:    177560
001164  177562                  TKB:    177562
001166  177564                  TPS:    177564          ;IF THIS AND NEXT LOC ARE CHANGED,
001170  177566                  TPB:    177566          ;CHANGE LOC.S LPWK1 AND LPWK2 IF PRESENT
001172  177514                  LPS:    177514
001174  177516                  LPB:    177516
001176  177570                  SR:     177570                  ;ADDRESS OF THE SWITCH REGISTER
001200  123456                  RANNUM: 123456          ;RANDOM NUMBER IS CALC'ED AND PASSED HERE
001202  022333                  RANWRK: 22333           ;ALSO USED BY RAND.
001204  000000                  MEMVA:  OPEN
001206  000000                  MEMPA:  OPEN
001210  000000                  MEMEA:  OPEN
001212  000000                  MEM12:  OPEN
001214  000000                  MEMSBE: OPEN
001216  000000                  MEMWAS: OPEN
001220  000000                  LOCORE: OPEN                    ;STARTING ADDR OF FREE CORE (MULTIPLE OF 1000 OCTAL)
001222  000000                  ZERO:   0                       ;LOCORE EA BITS
001224  000000                  LCOR12: OPEN                    ;TOP 12 BITS OF LOW CORE ADDRESS
001226  000000                  HCOR12: OPEN                    ;CONTAINS ST. ADDR OF HIGHEST BUFFER,
                                                                ;(TOP 12 BITS ONLY)
001230  000000                  HCORSV: OPEN
001232  001534                  WBUF:   START                   ;CONTAINS CURRENT WRITE BUFFER ADDR,
001234  000000                  WBFEA:  OPEN                    ;WRITE BUFFER EA BITS (SHIFTED TO POS 4,5)
001236  000000                  WBUF12: OPEN                    ;TOP 12 BITS OF WBUF ADDRESS IF USING ROTATION
001240  000000                  OFFMX:  OPEN                    ;TOP ALLOWABLE PROGRAM OFFSET
001242  000000                  OFFSET: OPEN                    ;CURRENT PROGRAM OFFSET
001244  000000                  OFFSTD: OPEN                    ;DESTINATION OFFSET
001246  000000                  OFFSTM: OPEN                    ;SAVE LOCATION FOR DESTINATION OFFSET
                                                                ;DURING MOVE
001250  000000                  MOVBNK: OPEN                    ;BANK BEING MOVED (SOURCE ADDRESS)
001252  000000                  MAPRET: OPEN                    ;RETURN ADDRESS FROM MVCODE ROUTINE
001254  000024                  ERRLIM: 20.
001256  012000                  MXWBF:  AMODNM-START/2&177000   ;MAX SIZE OF WRITE BUFFER, VARIES
                                                                ;IF BUFFER ROTATION IS ENABLED
001260  000000                  TTYBYT: OPEN
001262  000000                  CXRET:  OPEN                    ;CTRLX RETURN ADDRESS
001264  000000                  SYSCNT: OPEN                    ;TOTAL NUMBER OF SYSTEM ERRORS
001266  000000                  PWRCNT: OPEN                    ;TOTAL NUMBER OF POWER FAILS
001270  000000                  BKPTR:  OPEN                    ;QUEUE POINTER INITIALIZATION FOR RUNNING BKMODS
001272  000000                  KTSAV:  0               ;USED FOR SECOND PASS IN CHAIN MODE TO
001274  000000                  HCR12A: 0               ;RESTORE TRUE STATE OF SYSTEM
001276  177600                  PDRTAB: 177600
001300  172200                          172200
001302  172300                  PDREND: 172300
001304  000000                  KTPRES: OPEN                    ;0= INHIBITED OR NOT PRESENT (LEAVE AS FULL
                                                                ;WORD FOR EASY MODIFICATION AND TO GUARANTEE
                                                                ;THAT FILCNT AND FILLER ARE IN SAME WORD)
001306      014                 FILCNT: .BYTE   12.             ;FILCNT AND FILLER MUST BE IN SAME WORD
001307      000                 FILLER: .BYTE   0
001310      000                 SYSERI: .BYTE   OPEN
001311      000                 RMODE:  .BYTE   OPEN            ;RUN MODE INDICATOR
001312      000                 CHN:    .BYTE   OPEN
001313      000                 ROTI:   .BYTE   OPEN
001314      000                 ROTL:   .BYTE   OPEN            ;WRITE BUFFER ROTATION LOCKED FLAG
```



```
001315      000                 RELOC:  .BYTE   OPEN            ;RELOCATION ALLOWED FLAG
001316  000000                  ROTCNT: OPEN                    ;ROTATION COUNTER USED WITH ROTL
001320  000100                  ROTNUM: 100                     ;ROTATION CONSTANT USED WITH ROTCNT
001322  000400                  ROTSIZ: 400                     ;ROTATE PROGRAM BY 4K
001324      000                 FILLID: .BYTE   OPEN            ;FILL INDICATOR,
001325      000                 MOVING: .BYTE   OPEN            ;CODE BEING RELOCATED
001326      000                 RUNRFL: .BYTE   OPEN
001327      000                 LOCK:   .BYTE   OPEN            ;PREVENT ROTATION OF CODE EXECUTION
001330      000                 MVDOWN: .BYTE   OPEN
001331      000                 RETRY:  .BYTE   OPEN            ;RETRY COUNT FOR CODE RELOCATION
001332      010                 QUECNT: .BYTE   10              ;# OF TIMES TO CHECK IOQUE BEFORE CHECKING
                                                                ;TYPEQ WHEN BOTH HAVE REQUESTS PENDING
001333      000                 MDL45:  .BYTE   OPEN            ;NON-ZERO IF CPU IS 11/45
001334      000                 $GWBFC: .BYTE   OPEN            ;WRITE BUFFER ROTATION COUNT
001335      000                 MEMERF: .BYTE   OPEN
001336      010                 BKTIME: .BYTE   10
001337      003                 ROTN:   .BYTE   3               ;# OF GWBUF CALLS (LESS ONE) NEEDED PER
                                                                ;ROTBUF CALL.
001340      000                 PPRES:  .BYTE   OPEN            ;PARITY PRESENT, 0=INHIBITED OR NONEXISTENT
001341      000                 CNT1:   .BYTE   OPEN
001342      000                 ATE:    .BYTE   OPEN
001343      000                 MFLAG:  .BYTE   OPEN
001344      000                 HNGFLG: .BYTE   OPEN            ;FLAG INDICATES CKHUNG ROUTINE IS BUSY
001345      000                 MOVFLG: .BYTE   OPEN            ;FLAG USED IN MVCODE ROUTINE
001346      000                 MSGFLG: .BYTE   OPEN            ;FLAG INDICATING MSGN OR MSGS CALL
001347      000                 DATFLG: .BYTE   OPEN            ;FLAG INDICATING CKDATA OR DATACK CALL
001350      012                 SYSLIM: .BYTE   10.             ;MAX. NUMBER OF SYSTEM ERRORS ALLOWED
001351      000                 TFLAG:  .BYTE   OPEN            ;TIMER FLAG INDICATING ERROR OR EOP
001352      000                 NSTOP:  .BYTE   OPEN            ;FLAG PREVENTING STOP FROM BEING SET
001353      000                 CLOCK:  .BYTE   OPEN            ;CLOCK FLAG
001354      000                 ALLBK:  .BYTE   OPEN            ;WHEN NON 0 MEANS RUN ALL BKMODS BEFORE RELOCATING
001356                                  .EVEN
001356  000000G                 CLOCKL: KW11L                   ;WILL BE 0 IF NO KW11-L CLOCK
001360  000000G                 CLOCKP: KW11P                   ;WILL BE 0 IF NO KW11-P CLOCK

001362  001604                  HNGTIM: 900.                    ;HUNG TIME LIMIT IN SECONDS
001364  000000G                 ALCLR:  LCLEAR                  ;SUBROUTINE ADDRESS IN L CLOCK MODULE
001366  000000G                 APCLR:  PCLEAR                  ;SUBROUTINE ADDR IN P CLOCK MODULE
                                                                ;WILL DROP ANY I/O MODULE INACTIVE FOR 15. MINS.
```

```
                        ;       *** DO NOT CHANGE THE SEQUENCE OF THE NEXT 9 WORDS *** !

                        ;REFER TO THE DOCUMENT FOR THE MONITOR CHECKSUM MODULE (BKBA)
                        ;FOR THE USE OF THE NEXT SIX WORDS, AND HOW THE TWO "OPEN" LOCATIONS
                        ;ARE DEFINED WHEN A NEW VERSION OF THE MONITOR IS RELEASED

001370  137577          CHKL2:  137577          ;LOW CHKSUM FOR PROC. WITH NO RTT INST.
001372  002641          CHKH2:  2641            ;HIGH CHKSUM
001374  001534          CHKSUM: .WORD   START            ;BEGIN ADDR OF MONITOR PURE AREA
001376  025106                  .WORD   MCASEE           ;END ADDR OF MONITOR PURE AREA
001400  137607          CHKL1:  137607          ;CONTAINS LOW CKSM OF MONITOR PURE AREA (PROC WITH RTT)
001402  002641          CHKH1:  2641            ;CONTAINS HI CKSM OF MONITOR PURE AREA
        001405                  .ODD
001405  000003          HOURS:  .BLKB   3               ;HOURS IN ASCII
001410  000000          MINITS: OPEN                    ;MINUTES IN ASCII
001412  000000          SEKNDS: OPEN                    ;SECONDS IN ASCII
001414  000000          RTIME:  OPEN                    ;TOTAL ELAPSED RUNTIME IN SECONDS
001416  000000          RTIMX:  OPEN                    ;HOLDS RUNTIME EXTENDED BITS
001420  000000          CLKADR: OPEN                    ;SAVE LOCATION FOR CLOCK ADDRESS
001422  000000          CLKHLD: OPEN                    ;SAVE LOCATION FOR CLOCK CSR
001424  000000          PTIME:  OPEN                    ;HOLDS MODULE PASS TIME IN SECONDS
001426  000000          MAXTIM: OPEN                    ;HOLDS MAX. ELAPSED TIME FOR CKHUNG ROUTINE
001430  000020          PARTAB: .BLKW   16.             ;TABLE OF PARITY REGISTER ADDRESSES
001470  000000          CPUERR: OPEN                    ;ADDRESS OF 11/60 CPU ERROR REG.
001472  000000          MEMERR: OPEN                    ;ADDRESS OF 11/60 MEMORY ERROR REG.
001474  000000          LGHOLD: OPEN                    ;HOLDS CONTENTS OF ERROR LOG REG. LAST READ/WRITTEN
001476  177746          CONTRL: 177746          ;CACHE CONTROL REG
001500  000000          KONTRL:         0               ;TEMP STOR FOR CACHE OPERATIONS

                                .EVEN
```



```
                        ;VARIABLE QUE CALL AREA- KEPT IN FRONT OF START SINCE IT IS IMPURE CODE

                        ;COMMON "QUE CALL" ROUTINE. DOES A DIRECT DISPATCH RATHER THAN
                        ;REQUEING DUE TO MULTIPLE POINTS CALLING THIS ROUTINE WHICH
                        ;MAY OVERLAP IF REQUEUED.

001502  012700  001512  COMQUE: MOV     #CADDR,R0
001506  000167  001304          JMP     IOQSVN
001512  000000          CADDR:  OPEN                    ;DESTINATION ADDR.
001514  000000          CSTART: OPEN                    ;MODULE START ADDR. (0 FOR MONITOR).

                        ;CHECKDATA ERROR CALL
001516  104421          CDERR:  DERRX$
001520  000000          CDERR1: OPEN                    ;ADDRESS OF CKDATA CALL
001522  000167  006736          JMP CDRET

                        ;TELETYPE ROUTINE PIRQ CALL
                        ;TTYLNK IS FREQUENTLY MODIFIED DURING EXECUTION

001526  000004          TTYSRV: PIRQ$
001530  007436          TTYLNK: TTSRV1
001532  000000                  0
```

```
                                        ;STARTUP SEQUENCE
                                        ;
001534  000005                  START:  RESET                   ;CLEAR THE WORLD.
001536  012706  027212                  MOV     #SPBOT,R6       ;SET UP STACK.
001542  012767  015460  176254          MOV     #PWRDN,PWRFV    ;SET UP POWER FAIL VECTOR.
001550  012746  000004                  MOV     #BUSEV,-(SP)    ;SAVE PRESENT BUS ERR TRAP
001554  012767  177570  177414          MOV     #HARDSR,SR      ;SET SR=HARDWARE SR
001562  012767  001576  176214          MOV     #3$,BUSEV       ;SET NEW BUS ERR TRAP
001570  005777  177402                  TST     @SR             ;IS IT HARDWARE SR?
001574  000404                          BR      4$              ;IF YES, RESTORE TRAP
001576  012767  001004  177372  3$:     MOV     #FAKESR,SR      ;ELSE SET UP SOFTWARE SR
001604  022626                  4$:     CMP     (SP)+,(SP)+             ;RESTORE STACK
001606  012667  176172                  MOV     (SP)+,BUSEV     ;RESTORE BUS ERR TRAP
001612  012767  026520  177450          MOV     #MODD-2,BKPTR   ;INITIALIZE TO START BKMODS WITH 1ST ONE
001620  005767  177374                  TST     LOCORE          ;DONE BUFFER SETUP?
001624  001054                          BNE     2$              ;BR IF YES.
001626  004567  025420                  JSR     R5,SETBUF       ;NO, DO IT.
                                        ;THE FOLLOWING IS USED TO WRITE ALL MEMORY WITH WHATEVER IS
                                        ;ALREADY THERE, JUST TO CLEAR ANY PARITY ERRORS PRESENT
001632  000005                          RESET                   ;TURN OFF KT
001634  105767  177444                  TSTB    KTPRES          ;KT?
001640  001430                          BEQ     14$             ;BR IF NO
001642  004767  010146                  JSR     PC,KTINIT       ;SET UP KT REG'S FOR NORMAL USE
001646  005067  170432                  CLR     KIPDR2          ;FORCE A KT TRAP WHEN LOOP OVERFLOWS
                                                                ;INTO NEXT PAGE
001652  012737  000001  177572          MOV     #1,@#SSR0       ;TURN ON KT
001660  012737  001736  000004          MOV     #176,@#4                ;EXIT POINT FOR END OF MEMORY
001666  012701  177600                  MOV     #-200,R1                ;SET BACK A PAGE FOR LOOP'S INC
001672  012737  001700  000250          MOV     #13$,@#250      ;SET UP KT TRAP TO BUMP TO NEXT PAGE
001700  062701  000200          13$:    ADD     #200,R1         ;BUMP TO NEXT PAGE
001704  010167  170432                  MOV     R1,KIPAR1       ;SET UP LOOP'S PAGE
001710  012706  001160                  MOV     #CCSRA,R6       ;GET RID OF TRAP JUNK ON STACK
                                                ;USING THIS AREA FOR STACK TEMPORARILY SINCE ITS
                                                ;IN THE SAME PAR SPACE
001714  012700  020000                  MOV     #20000,R0       ;SET R0 TO USE PAR 1
001720  000404                          BR      15$             ;GO DO A PAGE
001722  012737  001744  000004  14$:     MOV     #18$,@#4                ;SET UP EXIT FOR NON KT
001730  005000                          CLR     R0              ;START AT LOC 0
001732  011020                  15$:    MOV     (R0),(R0)+      ;WRITE A LOC WITH ITSELF
001734  000776                          BR 15$                  ;BR TILL KT OR 4 TRAP
001736  012737  000000  177572  17$:    MOV     #0,@#SSR0       ;TURN KT BACK OFF
001744  012706  027212  18$:    MOV     #SPBOT,R6       ;RESET STACK TO REAL AREA
001750  012737  015246  000004          MOV     #BUSERR,@#4     ;CORRECT TRAP VECTOR
                                        ;
001756  004767  013004          2$:     JSR     PC,CLRQUS       ;CLEAR QUEUES.
001762  105067  177323                  CLRB    RMODE
001766  112767  177777  177025          MOVB    #-1,MDXCTR      ;INITIALIZE COUNTER
001774  105067  177314                  CLRB    ROTL            ;MAKE SURE ROTATION IS NOT LOCKED
002000  104406  022672                  MSG$,TITLE              ;TYPE THE TITLE
002004  022767  001004  177164          CMP     #FAKESR,SR      ;DO WE HAVE A REAL SR?
002012  001002                          BNE     55$             ;BR IF YES
002014  104406  024633                  MSG$,NOSWR              ;ELSE TELL HIM WHERE IT IS
002020  004767  014714          55$:    JSR     PC,ROTCHK       ;CHECK IF ROTATE ALLOWED AND TYPE RANGE.
002024  004767  015022                  JSR     PC,SYSIZ        ;PRINT OUT THE SYSTEM SIZE
002030  004767  015514                  JSR     PC,KLOCK        ;CHECK FOR A CLOCK AND SET FLAGS
002034  105767  177244                  TSTB    KTPRES          ;KT11 ALLOWED?
```



```
002040  001410                          BEQ     START3          ;NO= BRANCH
002042  004767  007746          START2: JSR     PC,KTINIT       ;YES= INITIALIZE
002046  012737  000001  177572          MOV     #1,@#SSR0       ;TURN ON 18-BIT KT11
002054  104406  024615                  MSG$,KTONM              ;"KT11 ENABLED"
                
002060  000402                          BR      START4          ;CONTINUE
002062  104406  023776          START3: MSG$,NOKT               ;"NO KT11"
                
002066  004767  010722          START4: JSR     PC,PONN         ;GO TURN ON PARITY IF AVAILABLE
                
002072  105737  000041                  TSTB    @#41            ;ANY LOAD MEDIUM BITS SET ?
002076  001402                          BEQ     1$              ;NO, CONTINUE
002100  104406  023520                  MSG$,CLR40              ;" TO EXERCISE LOAD MEDIUM YOU MUST CLEAR LOC 40
002104  005067  177134          1$:     CLR     OFFSTD          ;MAKE SURE PROGRAM IS NOT PARTIALLY RELOCATED
002110  004767  017360                  JSR     PC,MVCODE
002114  004767  012646                  JSR     PC,CLRQUS       ;CLEAR QUEUES IN CASE MVCODE OVER WROTE THEM
002120  004767  012724                  JSR     PC,CLRVEC       ;INITIALIZE VECTOR AREA
002124  105767  177154                  TSTB    KTPRES          ;KT11 IN USE ?
002130  001002                          BNE     5$              ;YES, CONTINUE
002132  004767  022534                  JSR     PC,LOADWC       ;NO, LOAD WORST CASE PATTERN IN BUFFER
002136  104406  022737          5$:     MSG$,DOT                ;TYPE DOT.
002142  105767  177144                  TSTB    CHN             ;CHAIN MODE?
002146  001405                          BEQ     INPUT           ;NO= GO AWAIT KEYBOARD INPUT
002150  112777  000015  176660  6$:     MOVB    #15,@KBPTR      ;CLEAN OUT KEYBOARD BUFFER WITH A CR
002156  000167  006526                  JMP     RUN             ;YES=START RUNNING.
                
002162  012706  027212          INPUT:  MOV     #SPBOT,R6       ;RESET STACK.
002166  105067  177116                  CLRB    SYSERI          ;CLEAR SYSTEM ERROR INDICATOR.
002172  012767  026712  176636          MOV     #KBUF,KBPTR     ;INITIALIZE KYBD BUFFER POINTER.
002200  052777  000100  176754          BIS     #IE,@TKS        ;ENABLE KYBD INTERRUPTS.
002206  116767  177120  176613          MOVB    QUECNT,QUECTR
```

```
                                        ;QUEUE CHECK.  NOTE THAT IF IOBKID IS NEGATIVE, NO TYPE QUE REQUESTS WILL BE SERVICED

002214  105367  176607        QUETST: DECB    QUECTR          ;COUNT TO PREVENT BREAK CALLS FROM LOCKING OUT TYPING
002220  003004                        BGT     1$              ;ALSO PREVENTS HIGH SPEED INT DEVICES FROM
                                                                ;LOCKING OUT TYPE QUEUE AND RUN SERVICE
                                                                ;STARTUP OF MODULES
002222  116767  177104  176577        MOVB    QUECNT,QUECTR   ;RESET QUE COUNTER
002230  000405                        BR      2$
002232  105767  176550        1$:     TSTB    IOQUE           ;IO QUE REQUEST PENDING?
002236  001402                        BEQ     2$
002240  000167  000476                JMP     IOQSVC          ;GO SERVICE IF YES.
002244  005767  176610        2$:     TST     IOBKID          ;STARTING I/O MODULES?
002250  100422                        BMI     QTSTC           ;BR IF YES.
002252  105767  176531        3$:     TSTB    TYPQUE          ;TYPE REQUEST PENDING?
002256  001412                        BEQ     QTSTB           ;BR IF NOT.
002260  105767  176530                TSTB    TTYBSY          ;TTY BUSY?
002264  001002                        BNE     QTSTA           ;BR IF YES.
002266  000167  000674                JMP     TYPSVC          ;NO. GO SERVICE TYPE QUEUE.
002272  005367  176612        QTSTA:  DEC     TYPCTR          ;CHECK FOR TTY HUNG
002276  001002                        BNE     QTSTB
002300  000167  005132                JMP     TTSRV1          ;UNHANG TTY
002304  005767  176560        QTSTB:  TST     TRCPC           ;BACKGROUND MODULE PENDING?
002310  001402                        BEQ     QTSTC
002312  000167  000522                JMP     BKQSVC          ;JUMP IF YES
002316  105767  176767        QTSTC:  TSTB    RMODE           ;IN RUN MODE?
002322  001734                        BEQ     QUETST          ;BR IF NOT. GO CHECK AGAIN.
```



```
                                        ;RUN MODE SERVICE ROUTINE.
                                        ;NOTE THAT MODULE'S STACK POINTER IS REINITIALIZED BUT OTHER
                                        ;REGISTERS HAVE ANY PREVIOUS VALUES RESTORED AT START OR RESTART

002324  105767  176463        RUNSVC: TSTB    BRAKE           ;IS THE BRAKE ON?
002330  001331                        BNE     QUETST          ;BR IF YES. DO NOT INIT MORE MODULES.
002332  105767  177016                TSTB    ALLBK           ;RUN ALL BKMODS BEFORE RELOCATING ?
002336  001407                        BEQ     12$             ;NO, CONTINUE
002340  105767  176460                TSTB    STOP            ;WAITING TO RELOCATE ?
002344  001407                        BEQ     13$             ;NO, CONTINUE
002346  012767  040020  176504        MOV     #40020,IOBKID   ;START UP ONLY BKMOD'S
002354  000415                        BR      1$              ;CONTINUE

002356  105767  176442        12$:    TSTB    STOP            ;STOP SET?
002362  001314                        BNE     QUETST          ;YES, DON'T START ANY MORE MODULES
002364  105767  176426        13$:    TSTB    MODCTR          ;MODCTR =0?
002370  001007                        BNE     1$              ;BR IF NOT.
002372  105767  176714                TSTB    CHN             ;YES, ARE WE IN CHAIN MODE?
002376  100004                        BPL     1$              ;BR IF NOT.
002400  012767  040020  176452        MOV     #40020,IOBKID   ;SWITCH IT TO BACKGROUND AND PREVENT LOOPING
002406  000702                        BR      QUETST          ;THRU BACKGROUND MODULE SERIES MORE THAN ONCE
002410  062767  000002  176422  1$:   ADD     #2,MODPTR       ;POINT TO NEXT MODULE.
002416  017700  176416                MOV     @MODPTR,R0      ;MODULE ADDR TO R0.
002422  001476                        BEQ     7$              ;BR IF NO ADDR.
002424  016046  000020                MOV     STAT(R0),-(SP)  ;GET MODULE'S STATUS
002430  042716  016000                BIC     #BIT12!BIT11!BIT10,(SP) ;MASK SPECIAL INDICATORS
002434  022667  176420                CMP     (SP)+,IOBKID    ;CORRECT MODULE TYPE TO RUN?
002440  001136                        BNE     11$             ;NO- BRANCH
002442  005767  176574                TST     OFFSET          ;IF OFFSET 0, IGNORE BITS 10 +11
002446  001414                        BEQ     2$              ; CONTINUE
```

```
  603  002450  032760  004000  000020          BIT     #BIT11,STAT(R0) ;BIT11 SET?
  604  002456  001127                          BNE     11$             ;YES- DON'T RUN MODULE SINCE OFFSET NOT 0
  605  002460  032767  001777  176554          BIT     #1777,OFFSET    ;OFFSET 32K MULTIPLE?
  606
  607  002466  001404                          BEQ     2$              ;YES- IGNORE BIT 10
  608  002470  032760  002000  000020          BIT     #BIT10,STAT(R0) ;NO- BIT10 SET?
  609  002476  001117                          BNE     11$             ;YES- DON'T RUN MODULE
  610  002500  005767  176354          2$:     TST     IOBKID          ;BACKGROUND MODULE?
  611
  612
  613
  614  002504  100410                          BMI     3$              ;BR IF NOT.
  615  002506  112767  177777  176277          MOVB    #-1,BRAKE       ;YES, APPLY BRAKE.
  616  002514  016767  176320  176546          MOV     MODPTR,BKPTR    ;RECORD MODQ ADDRESS OF CURRENT BKMOD
  617  002522  004767  015314                  JSR     PC,BKTYM        ;RECORD START TIME OF THIS BKMOD
  618  002526  032760  010000  000020  3$:     BIT     #BIT12,STAT(R0) ;EXTENDED I/O MODULE?
  619  002534  001402                          BEQ     4$              ;NO- BRANCH
  620  002536  105267  176260                  INCB    MDXCNT          ;YES, COUNT IT
  621  002542  105060  000005          4$:     CLRB    XFLAG(R0)       ;CLEAR INDICATOR
  622  002546  016060  000024  000046          MOV     SPOINT(R0),SVR6(R0) ;SET UP MODULE SP POINTER.
  623  002554  105767  176546                  TSTB    RUNRFL
  624  002560  001004                          BNE     5$
  625  002562  016067  000022  176264  44$:     MOV     INIT(R0),DSTADR ;SET UP DESTINATION ADDR.
  626  002570  000407                          BR      6$
  627  002572  016067  000062  176254  5$:      MOV     RSTRT(R0),DSTADR
  628  002600  005760  000026                  TST     PSCNT(R0)       ;HAS MODULE BEEN THRU START YET?
  629  002604  001001                          BNE     6$              ;BR IF YES
  630  002606  000765                          BR      44$             ;ELSE GO BACK
  631  002610  105367  176202          6$:     DECB    MODCTR          ;DECR COUNT OF MODS INITED.
  632  002614  000167  000212                  JMP     IOQSVE          ;GO TO START MODULE
  633  002620  022767  040000  176232  7$:     CMP     #40000,IOBKID   ;FINISHED WITH SPECIAL BKMOD ?
  634  002626  001004                          BNE     8$              ;BR IF NOT.
  635  002630  012767  041000  176222          MOV     #41000,IOBKID   ;YES, SWITCH TO NBKMOD
  636  002636  000434                          BR      10$             ;CONTINUE
  637  002640  022767  041000  176212  8$:     CMP     #41000,IOBKID   ;FINISHED WITH NBKMODS ?
  638  002646  001013                          BNE     9$              ;NO. CONTINUE
  639  002650  012767  140000  176202          MOV     #140000,IOBKID  ;YES. SWITCH TO IOMOD.
  640  002656  112767  177777  176135          MOVB    #-1,MDXCTR      ;INITIALLY LOCK MDXCTR
  641
  642
  643
  644
  645  002664  105067  176132                  CLRB    MDXCNT          ;INITIALIZE COUNT
  646
  647
  648
  649
  650
  651  002670  105067  176456                  CLRB    NSTOP           ;ALLOW STOP FLAG TO BE SET
  652  002674  000415                          BR      10$             ; CONTINUE
  653  002676  005767  176156          9$:     TST     IOBKID          ;IF ALREADY BACKGROUND
  654  002702  100012                          BPL     10$             ;DON'T RESET MDXCTR
  655
  656
  657
  658
```



```
  659  002704  116767  176112  176107          MOVB    MDXCNT,MDXCTR   ;SETUP MDXCTR
  660  002712  012767  040020  176140          MOV     #40020,IOBKID   ;SWITCH TO BACKGROUND MODE
  661
  662
  663
  664
  665  002720  016767  176344  176112          MOV     BKPTR,MODPTR    ;CONTINUE WITH NEXT BKMOD
  666                                   ;       MOVB    #1,RUNRFL       ;SET RUN RESTART FLAG11/60          **********
  667  002726  000403                          BR      11$
  668  002730  012767  026520  176102  10$:     MOV     #MODQ-2,MODPTR  ;POINT TO MODULE TABLE START.
  669  002736  000167  177252          11$:    JMP     QUETST
```

```
                                        ;I/O QUE SERVICE ROUTINE.
                                        ;SERVICES PIRQs,QUEs, AND BREAKs CALLS
                                        ;NOTE THAT PIRQ SERVICE CAN OCCUR ON TOP OF THIS ROUTINE- SO
                                        ;LEVEL 7 PRIORITY IS REQUIRED IF COUNT IS DECREMENTED AFTER IOQ2
                                        ;IS MOVED

002742  026727  176130  030072  IOQSVC: CMP     IOQ2,#IOQLIM    ;REACHED LIMIT OF QUEUE?
002750  103403                          BLO     1$              ;BR IF NOT.
002752  012767  027252  176116          MOV     #IOQ,IOQ2       ;RESET IOQ2.
002760  105367  176022          1$:     DECB    IOQUE           ;DECREMENT REQUEST COUNT
002764  017700  176106                  MOV     @IOQ2,R0        ;GET PC+2 OF CALL.
002770  062767  000002  176100          ADD     #2,IOQ2         ;UPDATE IOQ2.
002776  010001                          MOV     R0,R1           ;SAVE PC+2 OF CALL
003000  014046                          MOV     -(R0),-(SP)     ;PUSH CALL ON STACK
003002  022726  104407                  CMP     #BREAK$,(SP)+   ;BREAK CALL?
003006  001002                          BNE     2$              ;NO, CONTINUE
003010  000167  000440                  JMP     BREAK.          ;YES, GO SERVICE IT
003014  005720                  2$:     TST     (R0)+

003016  012067  176032          IOQSVM: MOV     (R0)+,DSTADR    ;GET DESTINATION ADDR.
003022  012000                          MOV     (R0)+,R0                ;GET MODULE ADDR. IS IT 0?
003024  001002                          BNE     IOQSVE          ;BR IF NOT. IT'S A MODULE.
003026  000177  176022                  JMP     @DSTADR         ;GO DO MONITOR FUNCTION.
003032  116067  000020  176026  IOQSVE: MOVB    STAT(R0),RSTAT  ;GET RUN STATUS.
003040  032760  020000  000020  IOQSVA: BIT     #BIT13,STAT(R0) ;MODULE STOPPED?
003046  001402                          BEQ     1$              ;NO, CONTINUE
003050  000167  177140                  JMP     QUETST          ;YES, FORGET IT
003054  032760  040000  000020  1$:     BIT     #BIT14,STAT(R0) ;IS MODULE SELECTED ?
003062  001002                          BNE     2$              ;YES, CONTINUE
003064  000167  177124                  JMP     QUETST          ;NO, FORGET IT
003070  010067  176042          2$:     MOV     R0,MODADR       ;SAVE R0. (MODULE ADDR).
003074  010667  176052                  MOV     R6,SPSAV        ;SAVE MONITOR STACK TOO!.
003100  062700  000050                  ADD     #SVR6+2,R0      ;RESTORE MODULE'S REGS.
003104  014006                          MOV     -(R0),R6                ;STARTING WITH STACK POINTER.
003106  014005          IOQSVD: MOV     -(R0),R5
003110  014004                          MOV     -(R0),R4
003112  014003                          MOV     -(R0),R3
003114  014002                          MOV     -(R0),R2
003116  014001                          MOV     -(R0),R1
003120  014000                          MOV     -(R0),R0
003122  016746  175740          IOQSVB: MOV     RSTAT,-(SP)     ;LOAD RUN STATUS.
003126  105066  000001                  CLRB    1(SP)           ;KERNEL, 1ST REGISTER SET
003132  016746  175716                  MOV     DSTADR,-(SP)    ;LOAD DESTINATION ADDR.
003136  000006          RTT1:   RTT                             ;GO TO DESTINATION.
```



```
                                        ;BACKGROUND QUEUE SERVICED HERE.

003140  016767  175724  175706  BKQSVC: MOV     TRCPC,DSTADR    ;SET UP DESTINATION ADDR.
003146  116767  175720  175712          MOVB    TRCPSW,RSTAT    ;SET UP RUN STATUS.
003154  017700  175660                  MOV     @MODPTR,R0      ;MODULE START ADDR TO R0.
003160  005067  175704                  CLR     TRCPC           ;CLEAR BK MODULE WAITING INDICATOR.
                                                                ;IF ERROR OCCURS, WILL STOP MODULE FROM FURTHER
                                                                ;EXECUTION UNTIL ERROR IS REPORTED
003164  000725                          BR      IOQSVA          ;GO GET GOING.
```

```
  723                                   ;TYPE QUE SERVICE ROUTINE.
  724                                   ;REMOVES PC+2 OF CALL FROM Q, PASSING IT IN R1
  725
  726  003166  026727  175710  031222  TYPSVC: CMP     TYPQ2,#TYPLIM   ;REACHED UPPER END OF QUEUE?
  727  003174  103403                          BLO     1$              ;BR IF NOT.
  728  003176  012767  030072  175676          MOV     #TYPEQ,TYPQ2    ;YES, RESET TYPQ2.
  729  003204  105267  175604          1$:     INCB    TTYBSY          ;INDICATE TTY BUSY.
  730  003210  017701  175666                  MOV     @TYPQ2,R1       ;GET PC+2 OF CALL.
  731  003214  062767  000002  175660          ADD     #2,TYPQ2        ;UPDATE TYPQ2.
  732  003222  105367  175561                  DECB    TYPQUE          ;DECREMENT REQUEST COUNT.
  733  003226  010146                          MOV     R1,-(SP)                ;PUSH PC+2 OF CALL
  734  003230  005741                          TST     -(R1)           ;POINT TO CALL.
  735  003232  010167  175610                  MOV     R1,NUMBER       ;SAVE ADDRESS OF CALL
  736  003236  011101                          MOV     (R1),R1         ;GET CALL
  737  003240  006301                          ASL     R1              ;TIMES 2.
  738  003242  016101  172244                  MOV     TYPTAB-TRP2-4(R1),R1    ;FORM SERVICE ADDR.
  739  003246  000201                          RTS     R1              ;GO TO IT, RESTORE P1.
  740
  741  003250  005310                  TYPTAB: .WORD   PASEND
  742  003252  005014  004042  004034          .WORD   ENDSVC,ERRSVC,ERSVC1
  743  003260  003426  000001  004024          .WORD   MSG,,1,ERSVC2,MSGN.,1,1,1,1,MSGS.,1,1,DERRX.,MSGD.
  744  003266  003322  000001  000001
  745  003274  000001  000001  003312
  746  003302  000001  000001  003562
  747  003310  003466
```



```
  748                                   ;MSGN. ROUTINE. SERVICES CALLS TO TYPE ASCII MESSAGES.
  749
  750  003312  112767  177777  176026  MSGS.:  MOVB    #-1,MSGFLG      ;SET MSGS CALL FLAG
  751  003320  000402                          BR      MSCONT          ;CONTINUE
  752  003322  105067  176020          MSGN.:  CLRB    MSGFLG          ;SET MSGN CALL FLAG
  753
  754  003326  012167  175532          MSCONT: MOV     (R1)+,TABADR    ;GET ASCII TABLE ADDR.
  755  003332  012167  176156                  MOV     (R1)+,CSTART    ;MODULE ADDR TO CSTART.
  756  003336  010167  176150                  MOV     R1,CADDR        ;RESUME ADDR TO CADDR.
  757  003342  016701  176146                  MOV     CSTART,R1       ;MODULE ADDR TO R1.
  758  003346  032777  020000  175622          BIT     #BIT13,@SP      ;INHIBIT ERROR PRINT?
  759  003354  001033                          BNE     MSG1.           ;BR IF YES.
  760  003356  105767  175764                  TSTB    MSGFLG          ;IS THIS A MSGS CALL ?
  761  003362  100410                          BMI     1$              ;YES, CONTINUE
  762  003364  004767  001300                  JSR     PC,ENDCOM       ;NO, DO COMMON STUFF.
  763  003370  004567  003754                  JSR     R5,TYPEN        ;TYPE
  764  003374  022744                          CRCR                    ;2 CARRAGE RETURNS
  765  003376  025721                          AEND                    ;COMMON HEADER.
  766  003400  025756                          $PASS                   ;PASS #.
  767  003402  000000                          0
  768  003404  016701  175454          1$:     MOV     TABADR,R1       ;TABLE ADDR TO R1.
  769  003410  012146                  2$:     MOV     (R1)+,-(SP)     ;GET MESSAGE ADDR.
  770  003412  022716  177777                  CMP     #-1,(SP)                ;TERMINATOR?
  771  003416  001412                          BEQ     MSG1.           ;BR IF YES, DONE.
  772  003420  004767  003742                  JSR     PC,TYPE         ;NO, TYPE MESSAGE.
  773  003424  000771                          BR      2$              ;GO DO IT AGAIN.
```

```
                                ;MSG$ CALL SERVICED HERE
                                
003426  012146                  MSG,:   MOV     (R1)+,-(SP)     ;ASCII MESSAGE ADDR TO STACK.
003430  010167  176056                  MOV     R1,CADDR        ;RESUME ADDR TO CADDR.
003434  005067  176054                  CLR     CSTART          ;INDICATE MONITOR QUE CALLING.
003440  004767  003722                  JSR     PC,TYPE         ;GO TYPE DESIRED MESSAGE.
003444  105067  175344          MSG1.:  CLRB    TTYBSY
003450  000167  176026                  JMP     COMQUE          ;GO QUEUE UP TO RESUME.
```



```
                                ;BREAK ROUTINE, SERVICES BREAK CALL.
                                
003454  012100                  BREAK,: MOV     (R1)+,R0                ;GET MODULE ADDR.
003456  010167  175372                  MOV     R1,DSTADR       ;GET DESTINATION ADDR.
003462  000167  177344                  JMP     IOQSVE          ;QUE UP TO RESUME
```

```
                                        ;MSGD0 SERVICE- TYPE DATA ERROR COUNT AND TRANSFER SIZE
                                        ;AND HALT MODULE IF NECESSARY
                                        
003466  010167  176020          MSGD.: MOV     R1,CADDR         ;SET UP RESUME ADDRESS
003472  004767  001254                 JSR     PC,TYPDAT        ;GET MODULE ADDRESS
003476  012601                         MOV     (SP)+,R1
003500  010167  176010                 MOV     R1,CSTART        ;SAVE ADDRESS
003504  004567  010124                 JSR     R5,FILLNM        ;GET MODULE NAME
003510  026424                         AERNM+1
003512  004767  001234                 JSR     PC,TYPDAT        ;SETUP ERROR COUNT (ASCII)
003516  004567  010650                 JSR     R5,BDCNV
003522  026436                         AERCT
003524  004767  001222                 JSR     PC,TYPDAT        ;SETUP TOTAL WORDS TRANSFERRED
003530  004567  010636                 JSR     R5,BDCNV
003534  026470                         AERTOT
003536  032777  020000  175432         BIT     #BIT13,@SR       ;INHIBIT ERROR PRINTOUT?
003544  001004                         BNE     MSGD1.           ;YES- BRANCH
003546  012746  026423                 MOV     #AERNM,-(SP)     ;NO- TYPE MESSAGE
003552  004767  003610                 JSR     PC,TYPE
003556  000167  001014          MSGD1.: JMP     ERSVND           ;CHECK FOR HALTING MODULE, AND RETURN IF NOT
```



```
                                        ;DERRX, ROUTINE, SERVICES EXTENDED DATA ERROR CALLS FROM THE MONITOR
                                        ;CHECKDATA ROUTINE, TYPES ERROR INFORMATION AND RETURNS
                                        
003562  112767  000001  175227  DERRX.: MOVB    #1,ERRIND
003570  005721                         TST     (R1)+            ;POINT TO RETURN LOCATION
003572  010167  175714                 MOV     R1,CADDR         ;SETUP RESUME ADDRESS
003576  004767  001150                 JSR     PC,TYPDAT        ;GET MODULE ADDRESS
003602  012601                         MOV     (SP)+,R1
003604  010167  175704                 MOV     R1,CSTART        ;AND SAVE IT
003610  004767  001136                 JSR     PC,TYPDAT
003614  012667  175226                 MOV     (SP)+,NUMBER     ;SETUP ADDRESS OF CKDATA CALL
003620  004767  001044                 JSR     PC,ENDCOM
003624  012702  000005                 MOV     #5,R2
003630  004767  001116          DERRX1: JSR     PC,TYPDAT
003634  005302                         DEC     R2
003636  001374                         BNE     DERRX1
003640  016146  000030                 MOV     ERCNT(R1),-(SP) ;ERROR COUNT TO STACK
003644  004567  010522                 JSR     R5,BDCNV         ;CONVERT ERROR COUNT TO DECIMAL
003650  026335                         AERNMB
003652  005046                         CLR     -(SP)
003654  005766  000002                 TST     2(SP)
003660  001402                         BEQ     1$
003662  012716  000060                 MOV     #60,(SP)         ;CSRA EA BITS
003666  004567  010260          1$:    JSR     R5,OACNVX        ;CSR ADDRESS- ASCII
003672  026205                         ADTE6
003674  012603                         MOV     (SP)+,R3         ;SAVE TABLE ADDRESS
003676  012667  175252                 MOV     (SP)+,ERRPOS     ;SAVE OFFSET COUNT
003702  006367  175246                 ASL     ERRPOS
003706  012346                         MOV     (R3)+,-(SP)      ;SETUP READ BUFFER PA
003710  012346                         MOV     (R3)+,-(SP)      ;GET EA BITS
003712  066766  175236  000002         ADD     ERRPOS,2(SP)
003720  103002                         BCC     2$
003722  062716  000020                 ADD     #20,(SP)
003726  005723                  2$:    TST     (R3)+            ;SKIP SIZE
003730  004567  010216                 JSR     R5,OACNVX        ;GET PA OF RDADR
003734  026265                         ADTE4
003736  012346                         MOV     (3)+,-(SP)       ;GET WRITE PA
003740  012346                         MOV     (3)+,-(SP)       ;GET EA BITS
003742  066766  175206  000002         ADD     ERRPOS,2(SP)
003750  103002                         BCC     3$
003752  062716  000020                 ADD     #20,(SP)
003756  004567  010170          3$:    JSR     R5,OACNVX        ;GET ASCII OF WRADR
003762  026250                         ADTE5
003764  004567  010150                 JSR     R5,OACNV         ;CONVERT S/B TO ASCII
003770  026220                         ADTE3
003772  004567  010142                 JSR     R5,OACNV         ;CONVERT WAS TO ASCII
003776  026233                         ADTE2
004000  016746  175150                 MOV     ERRPOS,-(SP)     ;ERROR POSITION IN BUFFER
004004  000241                         CLC                      ;CORRECT BACK FROM BYTE TO WORD PTR
004006  006216                         ASR     (SP)
004010  005216                         INC     (SP)             ;MAKE 1ST WORD #1
004012  004567  010354                 JSR     R5,BDCNV         ;GET DECIMAL
004016  026317                         ADTE7
004020  000167  000306                 JMP     ERSVX            ;GO TO COMMON ROUTINE
```

```
                                        ;ERROR CALLS ARE SERVICED HERE.
                                        ;R1 AT ENTRY CONTAINS PC+2 OF CALL

004024  112767  000003  174765  ERSVC2: MOVB    #3,ERRIND       ;INDICATE ERRORN CALL.
004032  000406                  ERSVC1: BR      ERSVCA
004034  105067  174757          ERSVC1: CLRB    ERRIND          ;INDICATE DATA ERROR.
004040  000403                  ERSVCA: BR      ERSVCA
004042  112767  000002  174747  ERRSVC: MOVB    #2,ERRIND       ;INDICATE "NORMAL" ERROR.
004050  012167  175440          ERSVCA: MOV     (R1)+,CSTART    ;SAVE START ADDR OF MODULE.
004054  122767  000003  174735          CMPB    #3,ERRIND               ;ERRORN CALL?
004062  001002                          BNE     1$              ;BR IF NOT
004064  012167  174774                  MOV     (R1)+,TABADR    ;SAVE TABLE ADDR.
004070  010167  175416          1$:     MOV     R1,CADDR        ;RESUME ADDR TO PSEUDB.
004074  016701  175414                  MOV     CSTART,R1       ;GET BACK START ADDR.
004100  004767  000564                  JSR     PC,ENDCOM       ;DO COMMON STUFF.
004104  005261  000030                  INC     ERCNT(R1)       ;INCREMENT MODULE'S ERROR COUNT.
004110  001002                          BNE     ERSVCB          ;BR IF RESULT NOT 0.
004112  005161  000030                  COM     ERCNT(R1)       ;RESET COUNT TO -1.
004116  012702  000005          ERSVCB: MOV     #5,R2           ;GET TYPE DATA FROM QUEUE TO STACK.
004122  004767  000624          ERSVCC: JSR     PC,TYPDAT       ;DO IT.
004126  005302                          DEC     R2              ;DONE?
004130  001374                          BNE     ERSVCC          ;BR IF NOT.
004132  016146  000030                  MOV     ERCNT(R1),-(SP) ;ERROR COUNT TO STACK.
004136  004567  010230                  JSR     R5,BDCNV        ;CONVERT ERROR COUNT TO DECIMAL.
004142  026335                          AERNMB
004144  105767  174647                  TSTB    ERRIND          ;DATA ERROR?
004150  001047                          BNE     2$              ;BR IF NOT.
004152  004567  007762                  JSR     R5,OACNV        ;CONVERT WAS TO ASCII.
004156  026233                          ADTE2
004160  004567  007754                  JSR     R5,OACNV        ;CONVERT S/B TO ASCII,RESET TYPQ2.
004164  026220                          ADTE3
004166  012667  174654                  MOV     (SP)+,NUMBER    ;GET ADDR OF LOCATION READ
004172  104413  001046                  GETPA$,NUMBER
004176  016746  174646                  MOV     NUMBPA,-(SP)
004202  016746  174644                  MOV     NUMBEA,-(SP)
004206  004567  007740                  JSR     R5,OACNVX       ;CONVERT WASADR TO ASCII.
004212  026265                          ADTE4
004214  012667  174626                  MOV     (SP)+,NUMBER
004220  104413  001046                  GETPA$,NUMBER
004224  016746  174620                  MOV     NUMBPA,-(SP)
004230  016746  174616                  MOV     NUMBEA,-(SP)
004234  004567  007712                  JSR     R5,OACNVX       ;CONVERT SBADR TO ASCII.
004240  026250                          ADTE5
004242  005046                          CLR     -(SP)           ;0 ONTO STACK
004244  016667  000002  174706          MOV     2(SP),CCSRA     ;SAVE CSR ADDRESS
004252  001402                          BEQ     1$              ;IF 0, BRANCH
004254  012716  000060                  MOV     #60,(SP)        ;SETUP EA BITS
004260  004567  007666          1$:     JSR     R5,OACNVX       ;CONVERT CSR ADDR TO ASCII.
004264  026205                          ADTE6
004266  000421                          BR      ERSVX
004270  022626          2$:     CMP     (SP)+,(SP)+             ;SKIP WAS AND S/B.
004272  004567  007642                  JSR     R5,OACNV        ;CONVERT STAT REG CONTENTS TO ASCII.
004276  026167                          ASTATC
004300  004567  007634                  JSR     R5,OACNV        ;CONVERT CSR CONTENTS TO ASCII.
004304  026152                          ACSRC
004306  005046                          CLR     -(SP)           ;0 ONTO STACK
```



```
004310  016667  000002  174642          MOV     2(SP),CCSRA     ;SAVE CSR ADDRESS
004316  001402                          BEQ     3$              ;BRANCH IF ADDRESS 0
004320  012716  000060                  MOV     #60,(SP)
004324  004567  007622          3$:     JSR     R5,OACNVX       ;CONVERT CSR ADDR TO ASCII.
004330  026136                          ACSRAC
004332  032777  020000  174636  ERSVX:  BIT     #BIT13,@SR      ;INHIBIT ERROR PRINT?
004340  001116                          BNE     ERSVND          ;BR IF YES.
004342  105767  175005                  TSTB    CLOCK           ;IS THERE A CLOCK AVAILABLE ?
004346  001414                          BEQ     7$              ;NO, CONTINUE
004350  004767  013476                  JSR     PC,ERRTYM       ;YES, SETUP TIME OF THE ERROR
004354  004567  002770                  JSR     R5,TYPEN        ;TYPE
004360  022744                          CRCR                    ;2 CARRAGE RETURNS
004362  026066                          TOTIM                   ;THE TOTAL TIME
004364  026111                          PSTIM                   ;THE PASS TIME
004366  025721                          AEND                    ;COMMON HEADER
004370  025756                          $PASS                   ;PASS NUMBER
004372  026327                          ERRNMB                  ;ERROR NUMBER
004374  000000                          0                       ;THAT'S ALL
004376  000407                          BR      8$              ;CONTINUE
004400  004567  002744          7$:     JSR     R5,TYPEN        ;TYPE
004404  022744                          CRCR                    ;2 CARRAGE RETURNS
004406  025721                          AEND                    ;COMMON HEADER.
004410  025756                          $PASS                   ;PASS #.
004412  026327                          ERRNMB                  ;ERROR #.
004414  000000                          0
004416  122767  000002  174373  8$:     CMPB    #2,ERRIND       ;DATA ERROR?
004424  003406                          BLE     1$              ;BR IF NOT.
004426  004567  002716                  JSR     R5,TYPEN        ;YES, TYPE
004432  026275                          ADTE4A                  ;DATA ERROR VALUES
004434  026177                          ADTERR
004436  000000                          0
004440  000404                          BR      2$
004442  012746  026130          1$:     MOV     #AERROR,-(SP)   ;TYPE ERROR MESSAGE.
004446  004767  002714                  JSR     PC,TYPE
004452  122767  000001  174337  2$:     CMPB    #1,ERRIND       ;EXTENDED DATA ERROR?
004460  001004                          BNE     3$              ;NO- BRANCH
004462  012746  026311                  MOV     #ADTEX,-(SP)    ;YES- OUTPUT POSITION IN BUFFER
004466  004767  002674                  JSR     PC,TYPE
004472  012746  022772          3$:     MOV     #ACRLF,-(SP)
004476  004767  002664                  JSR     PC,TYPE
004502  122767  000003  174307          CMPB    #3,ERRIND       ;ERRORN CALL?
004510  001032                          BNE     ERSVND          ;BR IF NOT.
004512  016702  174346                  MOV     TABADR,R2       ;TABLE ADDR TO R2.
004516  012703  000010          4$:     MOV     #8.,R3          ;WILL TYPE 8 VALUES PER LINE.
004522  022712  177777          5$:     CMP     #-1,(2)         ;TERMINATOR?
004526  001417                          BEQ     6$              ;BR IF YES. DONE.
004530  013246                          MOV     @(2)+,-(SP)     ;PUT VALUE IN STACK.
004532  004567  007402                  JSR     R5,OACNV        ;CONVERT IT TO OCTAL.
004536  025675                          HOLD6                   ;AND PUT IT HERE
004540  012746  025675                  MOV     #HOLD6,-(SP)    ;TYPE IT.
004544  004767  002616                  JSR     PC,TYPE
004550  005303                          DEC     R3              ;DONE 8 PER LINE?
004552  001363                          BNE     5$              ;BR IF NOT.
004554  012746  022772                  MOV     #ACRLF,-(SP)    ;OUTPUT CRLF.
004560  004767  002602                  JSR     PC,TYPE
004564  000754                          BR      4$              ;GO FOR MORE.
```

SEQ 0033

```
   973  004566  012746  022772          68$:    MOV     #ACRLF,-(SP)    ;OUTPUT CRLF.
   974  004572  004767  002570                  JSR     PC,TYPE
   975  004576  005777  174374          ERSVND: TST     @SR             ;HALT MODULE ON ERROR?
   976  004602  100410                          BMI     1$              ;BR IF YES.
   977  004604  026761  174444  000030          CMP     ERRLIM,ERCNT(R1)        ;ERROR COUNT 20 OR GREATER?
   978  004612  003010                          BGT     2$              ;BR IF NOT. CONTINUE MODULE EXECUTION.
   979  004614  032777  040000  174354          BIT     #BIT14,@SR      ;YES. HALT MODULE AFTER 20 ERRORS?
   980  004622  001004                          BNE     2$              ;NO= SKIP
   981  004624  005067  174330          1$:     CLR     CCSRA           ;CLEAR CCSRA
   982  004630  000167  000166                  JMP     ENDSVA          ;YES, GO HALT MODULE
   983  004634  105067  174154          2$:     CLRB    TTYBSY          ;QUE TO RESUME
   984  004640  005767  174314                  TST     CCSRA           ;IF NOT SETUP, DON'T RESTORE INTO MODULE
   985  004644  001407                          BEQ     3$
   986  004646  016700  174642                  MOV     CSTART,R0
   987  004652  016760  174302  000050          MOV     CCSRA,CSRA(R0)  ;RESTORE CSRA IN MODULE
   988  004660  005067  174274                  CLR     CCSRA
   989  004664  000167  174612          3$:     JMP     COMQUE
```


SEQ 0034

```
   990  004670  104413  001046          ENDCOM: GETPAS,NUMBER           ;NUMBER CONTAINS ADDRESS OF CALL
   991  004674  016746  174150                  MOV     NUMRPA,-(SP)
   992  004700  016746  174146                  MOV     NUMBEA,-(SP)
   993  004704  016746  174136                  MOV     NUMBER,-(SP)
   994  004710  160116                          SUB     R1,(SP)         ;COMPUTE ASSEMBLY PC.
   995  004712  004567  007222                  JSR     R5,OACNV        ;CONVERT ASSEMBLY PC TO ASCII.
   996  004716  025746                          AEND2
   997  004720  004567  007226                  JSR     R5,OACNVX       ;CONVERT PC TO ASCII.
   998  004724  025733                          AEND1
   999  004726  004567  006702                  JSR     R5,FILLNM       ;LET'S GET MODULE NAME.
  1000  004732  025721                          AEND                    ;STUFF AT AEND.
  1001  004734  016146  000026                  MOV     PSCNT(R1),-(SP) ;GET PASS COUNT.
  1002  004740  005216                          INC     (SP)            ;UP IT 1.
  1003  004742  004567  007424                  JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
  1004  004746  025766                          APASS
  1005  004750  000207                          RTS     PC              ;LET'S GET OUT.
```

```
1006                                    ;TYPDAT ROUTINE LOADS QUEUED DATA ONTO STACK.
1007
1008  004752  011646                    TYPDAT: MOV     (SP),-(SP)      ;SAVE EXIT ON STACK AGAIN.
1009  004754  026727  174122  031222            CMP     TYPQ2,#TYPLIM   ;REACHED END OF QUEUE?
1010  004762  103403                            BLO     1$              ;BR IF NOT.
1011  004764  012767  030072  174110            MOV     #TYPEQ,TYPQ2    ;YES. POINT TO START OF QUEUE.
1012  004772  017766  174104  000002  1$:       MOV     @TYPQ2,2(SP)    ;QUEUE DATA TO STACK.
1013  005000  062767  000002  174074            ADD     #2,TYPQ2        ;UPDATE QUEUE POINTER.
1014  005006  105367  173775                    DECB    TYPQUE          ;DECREMENT COUNT.
1015  005012  000207                            RTS     PC              ;EXIT. LEAVE DATA IN STACK.
```



```
1016                                    ;END CALL SERVICED HERE.
1017
1018  005014  011101                    ENDSVC: MOV     (R1),R1         ;GET START ADDR.
1019  005016  004767  177646                    JSR     PC,ENDCOM       ;DO COMMON STUFF.
1020  005022  052761  020000  000020    ENDSVA: BIS     #BIT13,STAT(R1) ;SET STOP BIT IN MODULE STAT.
1021  005030  105767  174317                    TSTB    CLOCK           ;CLOCK AVAILABLE ?
1022  005034  001413                            BEQ     1$              ;NO, CONTINUE
1023  005036  004767  013010                    JSR     PC,ERRTYM       ;YES, SETUP TIME OF DROP
1024  005042  004567  002302                    JSR     R5,TYPEN        ;TYPE
1025  005046  022744                            CRCR                    ;2 CARRAGE RETURNS
1026  005050  026066                            TOTIM                   ;TOTAL TIME
1027  005052  026111                            PSTIM                   ;MODULE PASS TIME
1028  005054  025721                            AEND                    ;COMMON HEADER
1029  005056  025775                            MODEND                  ;MODULE DROPPED
1030  005060  000000                            0                       ;THAT'S ALL
1031  005062  000406                            BR      2$              ;CONTINUE
1032
1033  005064  004567  002260            1$:     JSR     R5,TYPEN        ;TYPE
1034  005070  022744                            CRCR                    ;2 CARRAGE RETURNS
1035  005072  025721                            AEND                    ;COMMON HEADER.
1036  005074  025775                            MODEND                  ;END MESSAGE.
1037  005076  000000                            0
1038  005100  105067  173710            2$:     CLRB    TTYBSY          ;CLEAR TTY BUSY INDICATOR.
1039  005104  005761  000020    ENDSVE: TST     STAT(R1)                ;BACKGROUND MODULE?
1040  005110  100410                            BMI     1$              ;BR IF NOT.
1041  005112  105067  173675                    CLRB    BRAKE           ;RELEASE BRAKE.
1042  005116  105767  173702                    TSTB    STOP            ;STOP MODULES FOR RELOCATION?
1043  005122  001403                            BEQ     1$              ;NO, CONTINUE
1044  005124  105767  174224                    TSTB    ALLBK           ;RUN ALL BKMODS BEFORE RELOCATION ?
1045  005130  001413                            BEQ     ENDSVG          ;NO, CONTINUE
1046  005132  032761  010000  000020  1$:       BIT     #BIT12,STAT(R1) ;EXTENDED I/O MOD?
1047  005140  001402                            BEQ     6$              ;NO- BRANCH
1048  005142  105367  173654                    DECB    MDXCNT          ;YES, DEC EXTENDED MOD COUNT
1049  005146  105367  173643            6$:     DECB    MODCNT          ;DECR COUNT OF MODULES RUNNING.
1050  005152  105767  173646                    TSTB    STOP            ;RELOCATE CODE?
1051  005156  001445                            BEQ     ENDSVF          ;NO- BRANCH
1052  005160  126767  173631  173637  ENDSVG: CMPB    MODCNT,BKCNT    ;ALL I/O MODULES OFF?
1053  005166  003046                            BGT     ENDSVD          ;NO- BRANCH
1054  005170  105767  174160                    TSTB    ALLBK           ;RUN ALL BKMOD'S BEFORE RELOCATING ?
1055  005174  001404                            BEQ     1$              ;NO, CONTINUE
1056  005176  105767  173613                    TSTB    MODCNT          ;ALL MODULES DONE ?
1057  005202  001404                            BEQ     ENDSVB          ;YES, GO RELOCATE
1058  005204  000437                            BR      ENDSVD          ;NO, GO RUN THE REST OF THE MODULES
1059  005206  105767  173601            1$:     TSTB    BRAKE           ;YES- BKMOD OFF?
1060  005212  001034                            BNE     ENDSVD          ;NO- BRANCH (STOP BKMOD CLEANLY)
1061  005214  016767  174022  174022  ENDSVB: MOV     OFFSET,OFFSTD   ;YES- CALCULATE NEW OFFSET
1062  005222  066767  174074  174014          ADD     ROTSIZ,OFFSTD   ;BY STEPPING AHEAD 8K OR 16K
1063  005230  026767  174010  174002          CMP     OFFSTD,OFFMX    ;NEW ADDRESS IN RANGE?
1064  005236  101402                            BLOS    3$              ;YES- BRANCH
1065  005240  005067  174000                    CLR     OFFSTD          ;NO- START OVER IN BANK 0
1066  005244  004767  013064            3$:     JSR     PC,CLKOFF       ;MAKE SURE SYSTEM CLOCK IS OFF
1067  005250  004767  014220                    JSR     PC,MVCODE       ;MOVE PROGRAM
1068  005254  004767  013136                    JSR     PC,CLKON        ;TURN SYSTEM CLOCK BACK ON
1069  005260  112767  000001  174040            MOVB    #1,RUNRFL       ;SET FLAG TO INDICATE MODULE RESTART
1070  005266  000167  003642                    JMP     RUNRES          ;GO STARTUP AGAIN
1071  005272  105767  173517    ENDSVF: TSTB    MODCNT          ;ANY MODULES RUNNING?
```

```
 1072  005276  001002                        BNE     ENDSVD          ;YES- BRANCH
 1073  005300  000167  001544                JMP     CTRLCB          ;COUNT 0, TERMINATE RUN MODE.
 1074  005304  000167  174704        ENDSVD: JMP     QUETST          ;GO BACK TO SERVICE QUEUES.
```



```
 1075                                  ;PASEND ROUTINE, TYPES END OF PASS MESSAGE,
 1076                                  ;BACKGROUND MODULES ARE NOT ALLOWED TO MAKE MULTIPLE PASSES.
 1077
 1078  005310  011101                PASEND: MOV     (R1),R1         ;LOAD R1 WITH MODULE ADDRESS
 1079  005312  016167  000062  174172        MOV     RSTRT(R1),CADDR ;SAVE RESTART ADDRESS
 1080  005320  010167  174170                MOV     R1,CSTART       ;SAVE MODULE ADDRESS
 1081  005324  005261  000026                INC     PSCNT(R1)       ;INCREMENT PASS COUNT.
 1082  005330  105767  174017                TSTB    CLOCK           ;IS THERE A CLOCK AVAILABLE?
 1083  005334  001404                        BEQ     1$              ;NO, CONTINUE
 1084  005336  004767  013120                JSR     PC,CKHUNG       ;GO CHECK FOR ANY "HUNG" MODULES
 1085  005342  004767  012514                JSR     PC,PASTYM       ;SET UP RUNTIME, PASTIME INFORMATION
 1086  005346  032777  010000  173622  1$:   BIT     #BIT12,@SR      ;ENDPAS PRINTOUT?
 1087  005354  001432                        BEQ     PSENDA          ;BR IF NOT.
 1088  005356  004767  177306                JSR     PC,ENDCOM       ;DO COMMON STUFF.
 1089  005362  016146  000026                MOV     PSCNT(R1),-(SP) ;NOW GET IT.
 1090  005366  004567  007000                JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
 1091  005372  026016                        BPSCNT                  ;STUFF IT AT BPSCNT.
 1092  005374  105767  173753                TSTB    CLOCK           ;IS THERE A CLOCK AVAILABLE?
 1093  005400  001010                        BNE     2$              ;YES, CONTINUE
 1094  005402  004567  001742                JSR     R5,TYPEN        ;TYPE
 1095  005406  022744                        CRCR                    ;2 CARRAGE RETURNS
 1096  005410  025721                        AEND                    ;COMMON HEADER.
 1097  005412  026007                        APSEND                  ;ENDPAS AND COUNT.
 1098  005414  022742                        CR
 1099  005416  000000                        0
 1100  005420  000410                        BR      PSENDA          ;CONTINUE
 1101  005422  004567  001722          2$:   JSR     R5,TYPEN        ;TYPE
 1102  005426  022744                        CRCR                    ;2 CARRAGE RETURNS
 1103  005430  025721                        AEND                    ;COMMON HEADER
 1104  005432  026007                        APSEND                  ;END PASS AND COUNT
 1105  005434  026066                        TOTIM                   ;TOTAL TIME
 1106  005436  026111                        PSTIM                   ;PASS TIME
 1107  005440  000000                        0                       ;THAT'S ALL
 1108  005442  105067  173346          PSENDA: CLRB    TTYBSY          ;CLEAR TTY BUSY INDICATOR.
 1109  005446  105767  173640                TSTB    CHN             ;IN CHAIN MODE?
 1110  005452  100032                        BPL     10$             ;BR IF NOT
 1111                                  ;ELSE MAKE SURE EVERY MODULE KEEPS RUNNING TILL ALL HAVE COMPLETED
 1112                                  ;AN END OF PASS, THEN STOP ONE AT A TIME, THEN RETURN TO CHAIN MONITOR
 1113  005454  010046                        MOV     R0,-(SP)                ;SAVE R0
 1114  005456  010146                        MOV     R1,-(SP)                ;SAVE R1
 1115  005460  012700  026520                MOV     #MODQ-2,R0              ;GET LIST OF MODULE ADDRESS
 1116  005464  062700  000002          11$:  ADD     #2,R0           ;LOOK AT NEXT ENTRY
 1117  005470  011001                        MOV     (R0),R1         ;ANY MODULES LEFT?
 1118  005472  001416                        BEQ     12$             ;BR IF NOT, ALL DONE, DO NOT RESTART MODULES
 1119  005474  032761  040000  000020        BIT     #BIT14,STAT(R1) ;MODULE SELECTED?
 1120  005502  001770                        BEQ     11$             ;BR IF NOT
 1121  005504  032761  020000  000020        BIT     #BIT13,STAT(R1) ;WAS IT DROPPED?
 1122  005512  001364                        BNE     11$             ;BR IF YES
 1123  005514  005761  000026                TST     PSCNT(R1)       ;HAS HE DONE A PASS YET?
 1124  005520  001361                        BNE     11$             ;BR IF YES
 1125  005522  012601                        MOV     (SP)+,R1        ;RESTORE R
 1126  005524  012600                        MOV     (SP)+,R0        ;RESTORE R0
 1127  005526  000404                        BR      10$             ;GO LET THIS GUY DOING THE EOP RUN AGAIN
 1128  005530  012601                12$:    MOV     (SP)+,R1        ;RESTORE R1
 1129  005532  012600                        MOV     (SP)+,R0        ;RESTORE R0
 1130  005534  000167  177344                JMP     ENDSVE          ;START LETTING MODULES DIE OFF
```

```
 1131  005540  105767  173260          108$:   TSTB    STOP            ;STOP ALL MODULES TO ROTATE CODE?
 1132  005544  001402                          BEQ     14$     ;NO BR.
 1133  005546  000167  177332                  JMP     ENDSVE
 1134  005552  105267  173254          148$:   INCB    PASTOT          ;COUNT TOTAL PASSES AT PRESENT OFFSET
 1135  005556  105767  173531                  TSTB    ROTI            ;WRITE BUFFER ROTATION ENABLED?
 1136  005562  001403                          BEQ     1$              ;NO, TREAT LIKE ALL BKMODS RUNNING
 1137  005564  105767  173232                  TSTB    MDXCNT          ;ANY EXTENDED MODULES RUNNING?
 1138  005570  001021                          BNE     2$              ;YES- SKIP ALTERNATE RELOCATION METHOD
 1139  005572  126767  173234  173233  18$:    CMPB    PASTOT,PASREL   ;NO- UP TO RELOCATION NUMBER?
 1140  005600  103415                          BLO     2$              ;NO- BRANCH
 1141  005602  105767  173507                  TSTB    RELOC           ;YES- RELOCATION ALLOWED?
 1142  005606  001412                          BEQ     2$              ;NO- BRANCH
 1143  005610  105767  173513                  TSTB    LOCK            ;LOCK SET TO INHIBIT RELOCATION?
 1144  005614  001007                          BNE     2$              ;YES- BRANCH
 1145  005616  105767  173530                  TSTB    NSTOP           ;ALLOW STOP TO SET ?
 1146  005622  001004                          BNE     2$              ;NO, CONTINUE
 1147  005624  105267  173174                  INCB    STOP            ;YES, SET STOP
 1148  005630  000167  177250                  JMP     ENDSVE
 1149  005634  005761  000020          28$:    TST     STAT(R1)                ;BACKGROUND MODULE?
 1150  005640  100415                          BMI     PSENDB          ;BR IF NOT.
 1151  005642  105067  173145                  CLRB    BRAKE           ;RELEASE BRAKE.
 1152  005646  032761  001000  000020          BIT     #BIT9,STAT(R1)  ;IS IT AN NBKMOD ?
 1153  005654  001613                          BEQ     ENDSVD          ;NO, CONTINUE
 1154  005656  052761  020000  000020          BIS     #BIT13,STAT(R1) ;YES, SET IT'S STOP BIT
 1155  005664  105367  173135                  DECB    BKCNT           ;ONE LESS BKMOD RUNNING
 1156  005670  000167  177210                  JMP     ENDSVE          ;GO TAKE CARE OF OTHER STUFF
 1157  005674  000167  173602          PSENDB: JMP     COMQUE          ;GO TO COMMON QUE CALL.
 1158
 1159
```



```
 1160                                   ;TRACE TRAP ENTERS HERE.
 1161
 1162  005700  105367  173130          TRCI:   DECB    BKTIMR
 1163  005704  100401                          BMI     TRCIC
 1164  005706  000006                  TRCIA:  RTT
 1165  005710  116767  173422  173116  TRCIC:  MOVB    BKTIME,BKTIMR
 1166  005716  005767  173064                  TST     IOQUE           ;I/O OR TYPE QUE WAITING?
 1167  005722  001771                          BEQ     TRCIA           ;NO, EXIT.
 1168  005724  012667  173140          TRCIB:  MOV     (SP)+,TRCPC     ;SAVE MOD'S PC.
 1169  005730  012667  173136                  MOV     (SP)+,TRCPSW    ;SAVE MOD'S PSW.
 1170  005734  000401                          BR      EXIT1.
```

```
                                        ;EXIT CALL ENTERS HERE.
                                        
005736  022626                  EXIT.:  CMP     (SP)+,(SP)+
005740  010046                  EXIT1.: MOV     R0,-(SP)                ;SAVE R0 IN STACK.
005742  016700  173170                  MOV     MODADR,R0       ;MODULE ADDR TO R0.
005746  062700  000032                  ADD     #SVR0,R0        ;POINT TO MOD'S REG SAVE AREA.
005752  012620                          MOV     (SP)+,(R0)+     ;SAVE R0, (FROM STACK).
005754  010120                          MOV     R1,(R0)+                ;SAVE REMAINING REGS.
005756  010220                          MOV     R2,(R0)+
005760  010320                          MOV     R3,(R0)+
005762  010420                          MOV     R4,(R0)+
005764  010520                          MOV     R5,(R0)+
005766  010610                          MOV     R6,(R0)         ;SAVE MODULE STACK POINTER.
005770  016706  173156                  MOV     SPSAV,R6        ;RESTORE MONITOR STACK.
005774  005046                  EXIT2.: CLR     -(SP)           ;CLEAR PSW ON STACK
005776  012746  006004                  MOV     #1$,-(SP)       ;SET UP RETURN PC
006002  000002                          RTI                     ;CLEAR PSW AND CONTINUE
006004  000167  174204          1$:     JMP     QUETST
```



```
                                        ;TYPQ4. ROUTINE. LOADS MSGDs INFORMATION INTO TYPEQ

006010  004767  000242          TYPQ4.: JSR     PC,LDTYPQ       ;QUEUE PC+2 OF CALL
006014  016700  173116                  MOV     MODADR,R0       ;LOAD MODULE ADDRESS IN R0
006020  010046                          MOV     R0,-(SP)        ;AND SAVE IN QUEUE
006022  004767  000230                  JSR     PC,LDTYPQ
006026  016046  000060                  MOV     AWAS(R0),-(SP)  ;QUEUE DATA ERROR COUNT
006032  004767  000220                  JSR     PC,LDTYPQ
006036  016046  000056                  MOV     ASB(R0),-(SP)   ;QUEUE SIZE OF TRANSFER
006042  004767  000210                  JSR     PC,LDTYPQ
006046  005726                          POPSP                   ;REMOVE PS FROM STACK
006050  016706  173076                  MOV     SPSAV,R6        ;GET MONITOR STACK POINTER
006054  000747                          BR      EXIT2.          ;EXIT
```

```
1202                                    ;TYPQ3, ROUTINE QUEUES INFORMATION FOR DERRX CALLS
1203
1204  006056  011646                    TYPQ3,: MOV     (SP),-(SP)
1205  006060  004767  000172                    JSR     PC,LDTYPQ       ;QUEUE PC+2 OF CALL
1206  006064  016700  173046                    MOV     MODADR,R0       ;LOAD MODULE ADDRESS IN R0
1207  006070  010046                            MOV     R0,-(SP)        ;AND SAVE IN QUEUE
1208  006072  004767  000160                    JSR     PC,LDTYPQ
1209  006076  017616  000000                    MOV     @(SP),(SP)      ;QUEUE VIRTUAL ADDRESS OF CHECKDATA CALL
1210  006102  004767  000150                    JSR     PC,LDTYPQ
1211  006106  005726                            POPSP
1212  006110  062700  000062                    ADD     #AWAS+2,R0      ;GET ADDRESS OF INFORMATION
1213  006114  012701  000005                    MOV     #5,R1
1214  006120  000401                            BR      2$
1215  006122  005010                    1$:     CLR     (R0)            ;CLEAR DATA (EXCEPT LAST)
1216  006124  014046                    2$:     MOV     -(R0),-(SP)     ;GET DATA
1217  006126  004767  000124                    JSR     PC,LDTYPQ       ;QUEUE DATA
1218  006132  005301                            DEC     R1
1219  006134  001372                            BNE     1$
1220  006136  016700  172774                    MOV     MODADR,R0
1221  006142  010660  000046                    MOV     R6,SVR6(R0)
1222  006146  016706  173000                    MOV     SPSAV,R6
1223  006152  000167  177616                    JMP     EXIT2.
```



```
1224                                    ;TYPQ2, ROUTINE SERVICES ERROR, DATA ERROR, AND ERRN CALLS.
1225
1226  006156  011646                    TYPQ2,: MOV     (SP),-(SP)      ;SAVE PC+2 OF CALL AGAIN.
1227  006160  004767  000072                    JSR     PC,LDTYPQ       ;QUEUE UP PC+2 OF CALL.
1228  006164  010046                            MOV     R0,-(SP)                ;SAVE R0.
1229  006166  017600  000002                    MOV     @2(SP),R0       ;GET MODULE ADDR.
1230  006172  062700  000050            TYPQX:  ADD     #CSRA,R0        ;
1231  006176  012746  000005                    MOV     #5,-(SP)
1232  006202  012046                            MOV     (R0)+,-(SP)
1233  006204  000402                            BR      2$
1234  006206  011046                    1$:     MOV     (R0),-(SP)
1235  006210  005020                            CLR     (R0)+
1236  006212  004767  000040                    JSR     PC,LDTYPQ
1237  006216  005316                            DEC     (SP)
1238  006220  001372                            BNE     1$
1239  006222  005726                            POPSP
1240  006224  012600                            MOV     (SP)+,R0                ;RESTORE R0.
1241  006226  000167  177504                    JMP     EXIT.
```

SEQ 0045

```
                                ;TYPQ, ROUTINE, SERVICES END, ENDPAS, AND MSGN CALLS.

1244  006232  004767  000020    TYPQ,:  JSR     PC,LDTYPQ       ;QUEUE UP CALL.
1245  006236  005726                    POPSP
1246  006240  000167  177474            JMP     EXIT1,
```


SEQ 0046

```
                                ;TYPQ1, ROUTINE, SERVICES MSG CALL.

1249  006244  004767  000006    TYPQ1,: JSR     PC,LDTYPQ       ;QUEUE UP CALL.
1250  006250  005726                    POPSP
1251  006252  000167  177516            JMP     EXIT2,
```

```
                                        ;ROUTINE TO LOAD TYPE QUEUE, ENTERED VIA JSR PC.
006256  005046                 LDTYPQ: CLR     -(SP)           ;CLEAR PSW ON STACK
006260  012746  006266                 MOV     #3$,-(SP)       ;SETUP RETURN PC
006264  000002                         RTI                     ;CLEAR PSW AND CONTINUE
006266  105767  172515         3$:     TSTB    TYPQUE          ;REQUEST COUNT 0?
006272  001414                         BEQ     1$              ;BR IF YES.
006274  026767  172600  172600         CMP     TYPQ1,TYPQ2     ;NO, TYPQ1 AND TYPQ2 SAME?
006302  001010                         BNE     1$              ;BR IF NOT.
006304  012767  030072  172566         MOV     #TYPEQ,TYPQ1    ;RESET THE TYPE QUEUE
006312  104406  022774                 MSG$,QOVRFL     ;YES, QUEUE OFLO, CRASH SYSTEM
006316  104406  023746                 MSG$,HLT        ;'HALT'
006322  000000                         HALT
006324  026727  172550  031222 1$:     CMP     TYPQ1,#TYPLIM   ;REACHED HIGH LIMIT?
006332  001003                         BNE     2$              ;BR IF NOT.
006334  012767  030072  172536         MOV     #TYPEQ,TYPQ1    ;RESET TYPQ1.
006342  016677  000002  172530 2$:     MOV     2(SP),@TYPQ1    ;STORE ARGUMENT IN QUEUE
006350  105267  172433                 INCB    TYPQUE          ;UPDATE REQUEST COUNTS.
006354  062767  000002  172516         ADD     #2,TYPQ1        ;UPDATE TYPQ1.
006362  012616                         MOV     (SP)+,(SP)      ;
006364  000207                         RTS     PC              ;EXIT.
```



```
                                        ;ROUTINE TO LOAD BREAK$ CALL

006366  112766  177777  000003 LDBRK,:  MOVB    #-1,3(SP)       ;INDICATE BREAK$ CALL
006374  000402                         BR      LDIOQ

                                        ;ROUTINE TO LOAD QUE$ CALL.

006376  005066  000002         QUE,:   CLR     2(SP)           ;INDICATE QUE CALL.

                                        ;LDIOQ ROUTINE, MUST NOT DROP BELOW LEVEL OF INTERRUPTING DEVICE.

006402  012746  000340         LDIOQ:  MOV     #PRTY7,-(SP)    ;PUT NEW PSW ON STACK
006406  012746  006414                 MOV     #1$,-(SP)       ;SETUP RETURN PC
006412  000002                         RTI                     ;LOAD NEW PSW AND CONTINUE
006414  000403                 1$:     BR      .+10
006416  112766  000001  000002 PIRQ,:  MOVB    #1,2(SP)
006424  105767  172356                 TSTB    IOQUE           ;REQUEST COUNT 0?
006430  001414                         BEQ     1$              ;BR IF YES.
006432  026767  172436  172436         CMP     IOQ1,IOQ2       ;IOQ1 AND IOQ2 SAME?
006440  001010                         BNE     1$              ;BR IF NOT.
006442  012767  027252  172424         MOV     #IOQ,IOQ1       ;RESET IOQ1
006450  104406  022774                 MSG$,QOVRFL     ;QUE OFLO, CRASH SYSTEM
006454  104406  023746                 MSG$,HLT        ;'HALT'
006460  000000                         HALT
006462  026727  172406  030072 1$:     CMP     IOQ1,#IOQLIM    ;REACHED HIGH LIMIT?
006470  001003                         BNE     2$              ;BR IF NOT.
006472  012767  027252  172374         MOV     #IOQ,IOQ1       ;RESET IOQ1.
006500  012677  172370         2$:     MOV     (SP)+,@IOQ1     ;STORE PC+2 OF PENDING CALL.
006504  105267  172276                 INCB    IOQUE           ;UPDATE REQUEST COUNTS.
006510  062767  000002  172356         ADD     #2,IOQ1         ;UPDATE IOQ1.
006516  005716                         TST     (SP)            ;QUE CALL?
006520  001003                         BNE     3$              ;NO- BRANCH
006522  005726                         TST     (SP)+           ;YES, POP STACK
006524  000167  177244                 JMP     EXIT2,          ;EXIT
006530  126627  000001  177777 3$:     CMPB    1(SP),#-1       ;BREAK CALL?
006536  001003                         BNE     4$              ;NO- BRANCH
006540  005726                         TST     (SP)+           ;YES- POP STACK
006542  000167  177172                 JMP     EXIT1.
006546  005726                 4$:     TST     (SP)+           ;PIRQ- EXIT VIA RTI
006550  000002                         RTI
```

```
                                        ;KEYBOARD INTERRUPT SERVICE
                                        ;
006552  005077  172404          KBSRVC: CLR     @TKS            ;CLEAR INTERRUPT ENABLE.
006556  017746  172402                  MOV     @TKB,-(SP)      ;GET CHAR.
006562  042716  000200                  BIC     #200,(SP)       ;REMOVE EXTRA BITS.
006566  122716  000003                  CMPB    #3,(SP)         ;IS IT CTRL C?
006572  001524                          BEQ     CTRLCA          ;BR IF YES.
006574  005726                          POPSP                   ;REMOVE CHAR FROM STACK.
006576  000004  006604  000000          PIRQ$,1$,0              ;Q DEVICE SERVICE.
                                        ;
                                        ;KEYBOARD SERVICE.
                                        ;
006604  026727  172226  026752  1$:     CMP     KBPTR,#KBUF+KBUFL ;BUFFER EXCEEDED?
006612  001511                          BEQ     12$             ;BR IF YES.
006614  017746  172344                  MOV     @TKB,-(SP)      ;GET CHAR.
006620  042716  000200                  BIC     #200,(SP)       ;CLEAR PARITY BIT.
006624  121627  000177                  CMPB    (SP),#177       ;RUBOUT?
006630  001443                          BEQ     5$              ;BR IF YES.
006632  121627  000040                  CMPB    (SP),#40                ;SPACE? (IGNORE LEADING ONES).
006636  001460                          BEQ     8$              ;BR IF YES.
006640  121627  000015                  CMPB    (SP),#15                ;CR? (TIME TO QUIT?).
006644  001462                          BEQ     9$              ;BR IF YES.
006646  121627  000012                  CMPB    (SP),#12                ;LF? (TREAT LIKE CR).
006652  001457                          BEQ     9$              ;BR IF YES.
006654  121627  000060                  CMPB    (SP),#60                ;CHAR LESS THAN 60?
006660  103425                          BLO     10$             ;BR IF YES, IGNORE IT.
006662  121627  000141                  CMPB    (SP),#141       ;CHAR LESS THAN LOWER CASE A?
006666  103405                          BLO     2$              ;BR IF YES.
006670  121627  000172                  CMPB    (SP),#172       ;CHAR HIGHER THAN LOWER CASE Z?
006674  101017                          BHI     10$             ;BR IF YES.
006676  162716  000040                  SUB     #40,(SP)                ;LOWER CASE CHAR, MAKE IT UPPER CASE.
006702  111677  172130          2$:     MOVB    (SP),@KBPTR     ;STORE CHAR.
006706  005267  172124                  INC     KBPTR           ;UPDATE BUFFER POINTER.
006712  112667  172342          3$:     MOVB    (SP)+,TTYBYT    ;CHAR TO ECHO BUFFER.
006716  104406  001260                  MSG$,TTYBYT             ;ECHO THE CHARACTER.
006722  052777  000100  172232  4$:     BIS     #IE,@TKS        ;REENABLE KYBD INTERRUPTS.
006730  000167  173260                  JMP     QUETST
006734  005726                  10$:    POPSP
006736  000771                          BR      4$
                                        ;
                                        ;SERVICE DELETE/RUBOUT CODE.
                                        ;
006740  022767  026712  172070  5$:     CMP     #KBUF,KBPTR     ;BUFFER EMPTY?
006746  001405                          BEQ     6$              ;BR IF YES.
006750  005367  172062                  DEC     KBPTR           ;POINT TO PREVIOUS CHAR.
006754  117716  172056                  MOVB    @KBPTR,(SP)     ;GET PREVIOUS CHAR.
006760  000754                          BR      3$              ;GO ECHO IT.
006762  005726                  6$:     POPSP
006764  104406  022772                  MSG$,ACRLF              ;OUTPUT CRLF.
006770  012767  026712  172040  7$:     MOV     #KBUF,KBPTR     ;RESET KBUF POINTER.
006776  000751                          BP      4$
                                        ;
                                        ;SERVICE SPACE.
                                        ;
007000  022767  026712  172030  8$:     CMP     #KBUF,KBPTR     ;LEADING SPACE?
007006  001741                          BEQ     3$              ;BR IF YES (IGNORE IT).
```



```
007010  000734                          BR      2$              ;STORE IT.
                                        ;
                                        ;SERVICE CR AND LF.
                                        ;
007012  112677  172020          9$:     MOVB    (SP)+,@KBPTR    ;STORE CHAR.
007016  012767  026712  172012          MOV     #KBUF,KBPTR     ;POINT TO START OF KBUF.
007024  104406  022772                  MSG$,ACRLF              ;OUTPUT CRLF
007030  104401  010512  000000          QUE$,DECODE,0           ;QUE TO SERVICE COMMAND.
                                        ;
                                        ;SERVICE KBUF OVERFLOW.
                                        ;
007036  104406  023505          12$:    MSG$,KBOFLO             ;TYPE KBUF OFLO.
007042  000752                          BR      7$
```

```
                                        ;SERVICE CTRL C, ENDS RUN MODE ALSO,
                                        ;DROPS TO BANKS 0 AND 1 BEFORE RETURNING (IF KT11 IN USE)
                                        ;STACK POINTER SHOULD BE SETUP AHEAD OF CALL TO CTRLX TO PRESERVE
                                        ;NEEDED INFORMATION PRIOR TO REMAPPING TO 0, OR CAUSE REMAPPING TO
                                        ;CURRENT AREA AFTER
007044  012706  027212          CTRLCA: MOV     #SPBOT,SP       ;REINITIALIZE STACK POINTER
007050  004767  000130          CTRLCB: JSR     PC,CTRLX        ;CLEAR QUEUES, TYPE ^C
007054  005067  172164                  CLR     OFFSTD          ;GET PROGRAM BACK TO BANK 0
007060  004767  012410                  JSR     PC,MVCODE
007064  016767  012054  172074          MOV     LPWK1,TPS       ;RESTORE TTY
007072  016767  012050  172070          MOV     LPWK2,TPB
007100  105767  172205                  TSTB    RMODE           ;IN RUN MODE?
007104  001002                          BNE     1$              ;BR IF YES,
007106  000167  001544                  JMP     COMCO3          ;BACK TO KYBD ROUTINE,
007112  105067  172173          1$:     CLRB    RMODE           ;CLEAR RUN MODE INDICATOR
007116  105767  172170                  TSTB    CHN             ;IN CHAIN MODE?
007122  100426                          BMI     CTPLCD          ;BR IF YES, BYPASS SUMMARY AND RETURN TO
                                                                ;CHAIN MONITOR
007124  052777  000100  172030          BIS     #IE,@TKS        ;REENABLE KEYBOARD INTERRUPTS
007132  104406  022753                  MSG$,SUMARY             ;TYPE RUN END SUMMARY TITLE,
007136  105767  172211                  TSTB    CLOCK           ;IS THERE A CLOCK AVAILABLE?
007142  001406                          BEQ     2$              ;NO, CONTINUE
007144  004767  010506                  JSR     PC,RUNTYM       ;YES, SET UP RUNTIME INFORMATION
007150  104406  022747                  MSG$,AAT                ;"AT"
007154  104406  026066                  MSG$,TOTIM              ;PRINT THE TOTAL TIME
007160  104406  022742          2$:     MSG$,CR                 ;CARRIAGE RETURN
007164  004767  003144                  JSR     PC,DIRA         ;TYPE RUN SUMMARY,
007170  004767  007744                  JSR     PC,SYSNUM       ;TYPE NUMBER OF SYSTEM ERRORS
007174  004767  010024                  JSR     PC,PWRNUM       ;TYPE NUMBER OF POWER FAILS
007200  000167  006700          CTRLCD: JMP     CHNOUT          ;EXIT, OR RETURN TO KYBD RTN,
```



```
007204  012746  000340          CTRLX:  MOV     #PRTY7,-(SP)    ;SETUP NEW PSW ON STACK
007210  012746  007216                  MOV     #4$,-(SP)       ;PUT RETURN PC ON STACK
007214  000002                          RTI                     ;LOAD NEW PSW AND CONTINUE
007216  012767  000144  172270  4$:     MOV     #100.,CSTART    ;CLEAR QUEUES AND DELAY TOO,

007224  105767  172061                  TSTB    RMODE           ;IF NOT IN RMODE, SHORTEN WAIT
007230  001003                          BNE     1$
007232  012767  000010  172254          MOV     #10,CSTART
007240  004767  005522          1$:     JSR     PC,CLRQUS       ;CLEAR QUEUES,
007244  005367  172244                  DEC     CSTART          ;DONE?
007250  001373                          BNE     1$              ;BR IF NOT,
007252  011604                          MOV     (SP),R4         ;SAVE RETURN PC
007254  000005                          RESET
007256  010416                          MOV     R4,(SP)         ;RESTORE RETURN PC IN CURRENT BANK
007260  004767  005502                  JSR     PC,CLRQUS       ;CLEAR QUEUES IN BANK 0
007264  105767  172014                  TSTB    KTPRES
007270  001405                          BEQ     2$
007272  004767  002516                  JSR     PC,KTINIT
007276  012737  000001  177572          MOV     #1,@#SSPO
007304  132767  000001  172026  2$:     BITB    #RITO,PPRES             ;PARITY PRESENT ?
007312  001402                          BEQ     3$              ;NO, CONTINUE
007314  004767  002712                  JSR     PC,PAENBL       ;YES, GO TURN ON PARITY
007320  005046                  3$:     CLR     -(SP)           ;RETYPE NEW PSW ON STACK
007322  012746  007330                  MOV     #5$,-(SP)       ;PUT RETURN PC ON STACK
007326  000002                          RTI                     ;LOAD NEW PSW AND CONTINUE
007330  105267  171457          5$:     INCB    BRAKE           ;SET BRAKE TO PREVENT BACKGROUND MODULE FROM BEING START
007334  016767  171760  171754          MOV     ROTNUM,ROTCNT   ;RESET THE ROTATION COUNTER
007342  104406  022770                  MSG$,CTRLC              ;OUTPUT ^C
007346  000207                          RTS     PC              ;EXIT,
```

SEQ 0053

```
1442                                    ;SUBROUTINE TO TYPE MULTIPLE ASCII STRINGS, NOTE THAT THE PREVIOUS
1443                                    ;VALUE OF R5 IS NOT SAVED.
1444
1445  007350  005726                    TYPEN:  TST     (SP)+           ;REMOVE OLD R5 FROM STACK
1446  007352  012546                    TYPE1:  MOV     (5)+,-(SP)      ;GET ASCII ADDRESS.
1447  007354  001403                            BEQ     1$              ;BR IF 0 TERMINATOR.
1448  007356  004767  000004                   JSR     PC,TYPE         ;TYPE IT.
1449  007362  000773                            BR      TYPE1           ;GET MORE.
1450  007364  000205                    1$:     RTS     R5              ;RETURN.
```


SEQ 0054

```
1451                                    ;TYPE SUBROUTINE.
1452                                    ;TYPES ASCIZ STRING
1453                                    ;NOTE THAT IF RMODE IS SET BUT MODULES AREN'T YET BEING STARTED, BRAKE
1454                                    ;MUST BE SET TO PREVENT STARTNG MODULES WHILE TYPING
1455
1456  007366  012767  007436  172134  TYPE:   MOV     #TTSRV1,TTYLNK
1457  007374  012667  171506                  MOV     (SP)+,TYPRET    ;SAVE RETURN ADDRESS
1458  007400  012667  171500                  MOV     (SP)+,TYPADR    ;SAVE MESSAGE ADDRESS
1459  007404  010167  171530                  MOV     R1,TYPR1        ;SAVE REGISTERS 1-5
1460  007410  012701  001142                  MOV     #TYPR2,R1
1461  007414  010221                          MOV     R2,(R1)+
1462  007416  010321                          MOV     R3,(R1)+
1463  007420  010421                          MOV     R4,(R1)+
1464  007422  010521                          MOV     R5,(R1)+
1465  007424  012777  000100  171534          MOV     #100,@TPS       ;SET IE
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486  007432  000167  172556                  JMP     QUETST
1487  007436  117746  171442          TTSRV1: MOVB    @TYPADR,-(SP)   ;GET CHAR.
1488  007442  001015                          BNE     TYPEB           ;BR IF NOT TERMINATOR.
1489  007444  005726                          TST     (SP)+
1490  007446  012701  001142                  MOV     #TYPR2,R1       ;RESTORE REGISTERS 1-5
1491  007452  012102                          MOV     (R1)+,R2
1492  007454  012103                          MOV     (R1)+,R3
1493  007456  012104                          MOV     (R1)+,R4
1494  007460  012105                          MOV     (R1)+,R5
1495  007462  016701  171452                  MOV     TYPR1,R1
1496  007466  005077  171474                  CLR     @TPS            ;CLEAR IE
1497  007472  000177  171410                  JMP     @TYPRET
1498
1499
1500  007476  062767  000001  171400  TYPEB:  ADD     #1,TYPADR
1501  007504  122716  000045                  CMPB    #45,(SP)                ;IS IT %?
1502  007510  001037                          BNE     TYPEL           ;NO- BRANCH
1503  007512  112716  000015          TYPEC:  MOVB    #15,(SP)                ;OUTPUT CR.
1504  007516  012767  007526  172004          MOV     #1$,TTYLNK
1505  007524  000431                          BR      TYPEL
1506  007526  112746  000012          1$:     MOVB    #12,-(SP)       ;OUTPUT LF.
```

```
1507
1508
1509
1510  007532  012767  007542  171770          MOV     #28,TTYLNK
1511  007540  000423                          BR      TYPEL
1512  007542  116767  171540  171250  28$:    MOVB    FILCNT,FILCTR   ;GET FILL COUNT.
1513  007550  001002                          BNE     TYPEE           ;BR IF NOT 0.
1514  007552  105267  171242                  INCB    FILCTR          ;OOPS, MAKE IT A 1.
1515  007556  116746  171525          TYPEE:  MOVB    FILLER,-(SP)    ;OUTPUT FILLER.
1516  007562  012767  007572  171740  TYPED:  MOV     #38,TTYLNK
1517  007570  000407                          BR      TYPEL
1518  007572  105367  171222          38$:    DECB    FILCTR          ;DECREMENT FILL COUNTER.
1519  007576  001367                          BNE     TYPEE           ;BR IF NOT 0.
1520  007600  012767  007436  171722          MOV     #TTSRV1,TTYLNK
1521  007606  000713                          BR      TTSRV1
1522  007610  012677  171354          TYPEL:  MOV     (SP)+,@TPB
1523  007614  012767  010000  171266          MOV     #10000,TYPCTR
1524  007622  000167  172366                  JMP     QUETST
```



```
1525                                    ;ROUTINE TO SERVICE CKDATA CALL (CDATA8)
1526
1527  007626  112767  177777  171513  DATCK,: MOVB    #-1,DATFLG      ;SET FLAG FOR DATACK CALL
1528  007634  000402                          BR      CKCONT          ;CONTINUE
1529  007636  105067  171505          CDATA,: CLRB    DATFLG          ;SET FLAG FOR CKDATA CALL
1530
1531  007642  005067  171310          CKCONT: CLR     CDFLG
1532  007646  005067  171256                  CLR     DERRCT          ;INITIALIZE DATA ERROR COUNT
1533  007652  004467  005062                  JSR     R4,SAV04        ;TEMPORARILY SAVE REGS
1534  007656  016604  000012                  MOV     12(SP),R4       ;GET PC+2 OF CALL
1535  007662  010467  171632          CDATA1: MOV     R4,CDERR1       ;SETUP PC OF CALL
1536  007666  162767  000002  171624          SUB     #2,CDERR1
1537  007674  012401                          MOV     (R4)+,R1        ;GET BEGIN ADDRESS OF MODULE
1538  007676  012403                          MOV     (R4)+,R3        ;GET TABLE ADDRESS OF RBUFPA FROM MODULE
1539  007700  010361  000052                  MOV     R3,SBADR(R1)    ;SAVE IT
1540  007704  062701  000032                  ADD     #SVR0,R1        ;SAVE MODULE'S REGISTERS
1541  007710  012621                          MOV     (SP)+,(R1)+
1542  007712  012621                          MOV     (SP)+,(R1)+
1543  007714  012621                          MOV     (SP)+,(R1)+
1544  007716  012621                          MOV     (SP)+,(R1)+
1545  007720  012621                          MOV     (SP)+,(R1)+
1546  007722  010511                          MOV     R5,(R1)
1547  007724  022626                          CMP     (SP)+,(SP)+
1548  007726  016401  177774                  MOV     -4(R4),R1       ;RESTORE MODULE ADDR
1549  007732  012300                          MOV     (3)+,R0         ;GET READ BUFFER PA
1550  007734  012367  171164                  MOV     (3)+,MODREA     ;GET EA BITS ALSO
1551  007740  012305                          MOV     (3)+,R5         ;BUFFER WORD COUNT
1552  007742  016102  000074                  MOV     WBUFPA(R1),R2   ;GET WRITE BUFFER PA
1553  007746  016167  000076  171142          MOV     WBUFEA(R1),MODWEA  ;AND GET EA BITS
1554  007754  005003                          CLR     R3              ;CLEAR OUT REG, 3
1555  007756  005767  171174                  TST     CDFLG           ;ENTERED FROM DATA ERROR RETURN?
1556  007762  001421                          BEQ     30$             ;NO- BRANCH
1557  007764  012603                          MOV     (SP)+,R3        ;YES- GET COUNT
1558  007766  005203                          INC     R3
1559  007770  006303                          ASL     R3              ;CREATE OFFSET
1560  007772  060300                          ADD     R3,R0           ;GET ADDRESS OF CURRENT READ BUFFER WORD
1561  007774  103003                          BCC     .+10
1562  007776  062767  000020  171120          ADD     #20,MODREA
1563  010004  060302                          ADD     R3,R2           ;GET ADDRESS OF CURRENT WRITE BUFFER WORD
1564  010006  103003                          BCC     .+10
1565  010010  062767  000020  171100          ADD     #20,MODWEA
1566  010016  006203                          ASR     R3              ;RESTORE COUNT
1567  010020  160305                          SUB     R3,R5           ;CALCULATE WORDS LEFT
1568  010022  012667  171102                  MOV     (SP)+,DERRCT    ;RESTORE DATA ERROR COUNT
1569  010026  010067  171070          30$:    MOV     R0,MODRPA
1570  010032  010267  171056                  MOV     R2,MODWPA
1571  010036  004567  004560                  JSR     R5,TOP12S       ;GET 12 BIT WRITE BUFFER ADDRESS
1572  010042  001114                          MODWPA
1573  010044  004567  004552                  JSR     R5,TOP12S       ;GET 12 BIT READ BUFFER PA
1574  010050  001122                          MODRPA
1575  010052  105767  171226                  TSTB    KTPRES          ;USE KT11?
1576  010056  001443                          BEQ     1$              ;NO- BRANCH
1577  010060  012767  077406  167512          MOV     #77406,UIPDR0   ;MAP USER R/W
1578  010066  012767  077406  167506          MOV     #77406,UIPDR1
1579  010074  012767  077406  167502          MOV     #77406,UIPDR2
1580  010102  012767  077406  167506          MOV     #77406,UIPDR7
```

```
1581  010110  016767  162224  167522          MOV     KIPAR0,UIPAR0   ;EXECUTE IN PAGE 0
1582  010116  016767  170776  167516          MOV     MODW12,UIPAR1   ;PAGE 1 IS WRITE BUFFER
1583  010124  016767  170776  167512          MOV     MODR12,UIPAR2   ;PAGE 2 IS READ BUFFER
1584  010132  012767  007600  167516          MOV     #7600,UIPAR7    ;PAGE 7 IS EXTERNAL
1585  010140  042700  177700                  BIC     #177700,R0      ;SETUP READ BUFFER VA
1586  010144  052700  040000                  BIS     #40000,R0
1587  010150  042702  177700                  BIC     #177700,R2      ;SETUP WRITE BUFFER VA
1588  010154  052702  020000                  BIS     #20000,R2
1589  010160  052767  140000  167610          BIS     #UMODE,PSW      ;CHANGE TO USER
1590  010166  021012                  18$:    CMP     @R0,@R2         ;DATA OK?
1591  010170  001436                          BEQ     3$              ;YES, BRANCH
1592  010172  005267  170732                  INC     DERRCT          ;NO, COUNT DATA ERROR
1593  010176  026727  170726  000003          CMP     DERRCT,#3       ;MORE THAN 3 ERRORS?
1594  010204  003404                          BLE     2$              ;NO- REPORT ERROR
1595  010206  032777  002000  170762          BIT     #BIT10,@SR      ;YES- REPORT ALL?
1596  010214  001424                          BEQ     3$              ;NO- BRANCH
1597  010216  011067  170672          28$:    MOV     @R0,MODWPA      ;TEMPORARILY SAVE DATA
1598  010222  011267  170670                  MOV     @R2,MODWEA
1599  010226  104423                          RTTK$                   ;RETURN TO KERNEL
1600  010230  016761  170662  000056          MOV     MODWEA,ASB(R1)  ;SAVE EXPECTED
1601  010236  016761  170652  000060          MOV     MODWPA,AWAS(R1) ;SAVE ACTUAL
1602  010244  010361  000054                  MOV     R3,WASADR(R1)   ;SAVE COUNT
1603  010250  016746  170654                  MOV     DERRCT,-(SP)    ;SAVE DATA ERROR COUNT
1604  010254  010346                          MOV     R3,-(SP)        ;SAVE WORD COUNT ON STACK
1605  010256  016746  171236                  MOV     CDERR1,-(SP)    ;SAVE VA OF CKDATA CALL
1606  010262  000167  171230                  JMP     CDERR           ;GO TO ERROR CALL
1607  010266  005203                  38$:    INC     R3              ;COUNT WORDS CHECKED
1608  010270  005720                          TST     (R0)+
1609  010272  005722                          TST     (R2)+
1610  010274  105767  171004                  TSTB    KTPRES
1611  010300  001420                          BEQ     5$
1612  010302  032700  017777                  BIT     #17777,R0       ;REMAP IF NEEDED
1613  010306  001005                          BNE     4$
1614  010310  162700  020000                  SUB     #20000,R0
1615  010314  062767  000200  167322          ADD     #200,UIPAR2
1616  010322  032702  017777          48$:    BIT     #17777,R2
1617  010326  001005                          BNE     5$
1618  010330  162702  020000                  SUB     #20000,R2
1619  010334  062767  000200  167300          ADD     #200,UIPAR1
1620  010342  005305                  58$:    DEC     R5
1621  010344  001310                          BNE     1$
1622  010346  104423                          RTTK$                   ;RETURN TO KERNEL
1623  010350  005767  170554                  TST     DERRCT          ;ANY DATA ERRORS?
1624  010354  001003                          BNE     6$              ;YES, BRANCH
1625  010356  012467  170472                  MOV     (R4)+,DSTADR    ;GET THE RETURN ADDRESS
1626  010362  000423                          BR      7$              ;GO TO COMMON RETURN CODE
1627  010364  005261  000030          68$:    INC     ERCNT(R1)       ;DATA ERRORS OCCURRED- UP MODULE ERROR
1628                                                                  ;COUNT BY 1
1629  010370  001002                          BNE     .+6
1630  010372  005161  000030                  COM     ERCNT(R1)       ;SET TO -1 IF 0
1631  010376  016761  170526  000060          MOV     DERRCT,AWAS(R1) ;STORE INFORMATION IN MODULE
1632  010404  010361  000056                  MOV     R3,ASB(R1)      ;STORE SIZE IN ASB
1633  010410  010446                          MOV     R4,-(SP)        ;SAVE POINTER TO ERROR RETURN ADDRESS
1634  010412  010661  000046                  MOV     SP,SVR6(R1)     ;SAVE STACK POINTER
1635  010416  104422                          MSGD$                   ;REPORT TOTALS, HALT MODULE IF NEEDED
1636  010420  012604                          MOV     (SP)+,R4        ;GET ERROR RETURN POINTER
```



```
1637  010422  016701  170510                  MOV     MODADR,R1       ;GET MODULE START ADDRESS
1638  010426  012467  170422                  MOV     (R4)+,DSTADR    ;GET THE RETURN ADDRESS
1639  010432  105767  170711          78$:    TSTB    DATFLG          ;IS THIS A DATACK CALL ?
1640  010436  100002                          BPL     8$              ;NO, CONTINUE
1641  010440  016734  170464                  MOV     DERRCT,@(R4)+   ;YES, LOAD THE NUMBER OF DATA ERRORS
1642  010444  116167  000020  170414  88$:    MOVB    STAT(R1),PSTAT  ;SET UP FOR MODULE RETURN
1643  010452  010100                          MOV     R1,R0
1644  010454  062700  000046                  ADD     #SVR6,R0
1645  010460  000167  172422                  JMP     IOQSVD
1646
1647  010464  005746                  CDRET:  TST     -(SP)           ;PUSH STACK
1648  010466  004467  004246                  JSR     R4,SAV04
1649  010472  016604  000014                  MOV     14(SP),R4       ;GET VA OF CKDATA CALL
1650  010476  062704  000002                  ADD     #2,R4
1651  010502  005267  170450                  INC     CDFLG
1652  010506  000167  177150                  JMP     CDATA1
```

```
                                        ;COMMAND DECODER, CHECKS SYNTAX AGAINST COMMAND TABLE,
                                        ;ALL COMMANDS ARE CHECKED AGAINST THE QUOTES, NO ABBREVIATIONS ALLOWED

010512  105767  170272          DECODE: TSTB    SPCFLG          ;SPECIAL FLAG SET?
010516  001404                          BEQ     1$              ;BR IF NOT.
010520  005067  170264                  CLR     SPCFLG          ;YES, CLEAR IT.
010524  000177  170326                  JMP     @SRETRN         ;GO VIA SRETRN.
010530  105767  170555          1$:     TSTB    RMODE           ;IN RUN MODE?
010534  001403                          BEQ     2$              ;NO, CONTINUE
010536  104406  023131                  MSG$,INVCD1             ;'INVALID CMD, ONLY CNTL C IS OK'
010542  000441                          BR      COMCON          ;CONTINUE
010544  004467  004170          2$:     JSR     R4,SAV04        ;SAVE REGS 0-4.
010550  016700  170262                  MOV     KBPTR,R0        ;POINTER TO R0.
010554  012701  025106          GTOK:   MOV     #COMTAB,R1      ;DEVICE DECODING COMES HERE
010560  010046                  GTOKX:  MOV     R0,-(SP)        ;SAVE STRING POINTER
010562  122021                  GTOK1:  CMPB    (R0)+,(R1)+     ;MATCH THE CHARACTER?
010564  001407                          BEQ     GTOK2           ;YEP, SEE IF THE REST MATCH
010566  105721                          TSTB    (R1)+           ;NO, GO CHECK AGAINST NEXT
010570  001376                          BNE     .-2             ;BUT FIND WHERE NEXT ENTRY IS
010572  111102                          MOVB    (R1),R2         ;DISPLACEMENT TO NEXT ENTRY
010574  001422                          BEQ     COMCO1          ;NO SUCH THING,
010576  060201                  MORE:   ADD     R2,R1           ;POINT TO NEXT ENTRY
010600  011600                          MOV     (SP),R0         ;POINT TO BEGINNING OF INPUT
010602  000767                          BR      GTOK1           ;KEEP LOOKING
010604  105711                  GTOK2:  TSTB    (R1)            ;LAST CHR IN ENTRY?
010606  001365                          BNE     GTOK1           ;NO,CHECK THE OTHER CHR'S
010610  122126                          CMPB    (R1)+,(SP)+     ;RESET SP AND R1
010612  111102                          MOVB    (R1),R2         ;GET DISPATCH ADDRESS
010614  060201                          ADD     R2,R1           ;R1 POINTS TO NEXT ENTRY
010616  014167  170670                  MOV     -(R1),CADDR
010622  005067  170666                  CLR     CSTART          ;SET MODULE ADDR IN CSTART TO 0.
010626  010067  170204                  MOV     R0,KBPTR        ;SAVE CURRENT BUFFER POINTER.
010632  004767  004114                  JSR     PC,RST04        ;RESTORE REGS 0-4.
010636  000167  170640                  JMP     COMQUE          ;GO TO COMMON QUE CALL.
```



```
                                        ;COMMON RETURN FROM COMMAND EXECUTION AND RUN ROUTINE (INCLUDING
                                        ;RELOCATION AND INTERNAL RESTART)

010642  104406  023107          COMCO1: MSG$,INVCMD             ;TYPE INVALID COMMAND.
010646                          COMCON:
010646  005067  170136          COMCO2: CLR     SPCFLG          ;CLEAR SPECIAL FLAG AND
                                                                ;DIP COMMAND INDICATOR.
010652  105067  170446                  CLRB    FILLID          ;CLEAR FILL COMMAND INDICATOR.
010656  104406  022737          COMCO3: MSG$,DOT                ;TYPE DOT.
010662  000167  171274          COMCO4: JMP     INPUT           ;GO GET MORE INPUT.
```

```
1697                                    ;SPECIAL INPUT ROUTINE.
1698                                    ;CALLED BY MODIFY ROUTINE, WHEN WAITING FOR NEW VALUE TO BE INPUT.
1699
1700  010666  012667  170164  SINPUT: MOV     (SP)+,SRETRN    ;SAVE RETURN ADDR.
1701  010672  105267  170112          INCB    SPCFLG          ;SET SPECIAL FLAG.
1702  010676  000771                  BR      COMCO4
```



```
1703                                    ;RUN ROUTINE, STARTS EXERCISER EXECUTION.
1704                                    ;NOTE THAT "RUN" IS ENTERED ONLY FROM BANK 0, BUT "RUNRES" IS USED WHEN
1705                                    ;THE PROGRAM IS NOT IN BANK 0.
1706
1707  010700  112767  000001  170421  RUNL:   MOVB    #1,LOCK         ;SET LOCK TO PREVENT RELOCATING CODE EACH
1708                                                                    ;TIME WBUF IS CYCLED
1709  010706  000402                  BR      .+6
1710  010710  105067  170413  RUN:    CLRB    LOCK            ;CLEAR LOCK TO ALLOW CODE RELOCATION
1711  010714  005067  170324          CLR     OFFSTD
1712  010720  005067  170340          CLR     SYSCNT          ;ZERO SYSTEM ERROR COUNTER
1713  010724  005067  170336          CLR     PWRCNT          ;ZERO POWER FAIL COUNTER
1714  010730  132767  000001  170415  6$:     BITB    #BIT0,CLOCK     ;USING THE KW11-L CLOCK ?
1715  010736  001403                  BEQ     1$              ;NO, CONTINUE
1716  010740  004777  170420          JSR     PC,@ALCLR       ;YES, MAKE SURE CLOCK TABLE IS CLEARED
1717  010744  000406                  BR      2$              ;CONTINUE
1718  010746  132767  000002  170377  1$:     BITB    #BIT1,CLOCK     ;USING THE KW11-P  CLOCK ?
1719  010754  001402                  BEQ     2$              ;NO, CONTINUE
1720  010756  004777  170404          JSR     PC,@APCLR       ;YES, MAKE SURE CLOCK TABLE IS CLEARED
1721  010762  012767  042761  001540  2$:     MOV     #42761,SLDSB    ;RESTORE THESE  INSTRUCTIONS
1722  010770  012767  042761  001554          MOV     #42761,SLDSE    ;FOR A CHECK SUM MODULE
1723  010776  004767  002446          JSR     PC,GETADR
1724  011002  000436                  BR      5$              ;IF INVALID OR NO ADDRESS, FORCE 0
1725  011004  016701  170032          MOV     ADDRLO,R1       ;GET LOW ORDER BITS
1726  011010  016700  170030          MOV     ADDRHI,R0       ;GET ANY HIGH ORDER BITS
1727  011014  006200                  ASR     R0
1728  011016  006001                  ROR     R1
1729  011020  006200                  ASR     R0
1730  011022  006001                  ROR     R1
1731  011024  000241                  CLC
1732  011026  006001                  ROR     R1
1733  011030  006201                  ASR     R1
1734  011032  006201                  ASR     R1
1735  011034  006201                  ASR     R1
1736  011036  042701  000177          BIC     #177,R1         ;FORCE TO 4K BOUNDARY
1737  011042  001416                  BEQ     5$              ;IF NOW 0, SKIP REST (TO PREVENT ERR PRINTOUT
1738                                                            ;IF KTPRES WAS CLEARED)
1739  011044  020167  170170          CMP     R1,OFFMX
1740  011050  101403                  BLOS    3$              ;ADDRESS IS OK, CONTINUE
1741  011052  104406  023203          MSG$,INVCD2             ;'CANNOT GO THAT HIGH'
1742  011056  000673                  BR      COMCON          ;CONTINUE
1743  011060  105767  170220  3$:     TSTB    KTPRES          ;KT11 IN USE?
1744  011064  001003                  BNE     4$              ;YES, CONTINUE
1745  011066  104406  023241          MSG$,INVCD3             ;'NEED KT-11'
1746  011072  000665                  BR      COMCON          ;CONTINUE
1747  011074  010167  170144  4$:     MOV     R1,OFFSTD
1748  011100  112767  000001  170203  5$:     MOVB    #1,RMODE        ;MAKE SURE BANK 0 COPY OF RMODE IS
1749                                                            ;SET WHILE EXECUTING MVCODE
1750  011106  105067  170214          CLRB    RUNRFL
1751  011112  004767  010356          JSR     PC,MVCODE       ;MOVE PROGRAM TO NEW BANK AND GO THERE
1752  011116  105067  170167          CLRB    RMODE           ;CLEAR CURRENT COPY OF RMODE
1753  011122  012767  026520  170140          MOV     #MODG-2,BKPTR
1754  011130  004767  003714          JSR     PC,CLRVEC       ;INITIALIZE TRAPCATCHER IN VECTOR AREA
1755  011134  004767  003626  RUNRES: JSR     PC,CLRQUS       ;INCLUDES CLEARING BRAKE,MODCNT,IOBKID
1756  011140  132767  000002  170172          BITB    #BIT1,PPRES     ;IS IT AN 11/60 ?       **********************
1757  011146  001424                  BEQ     11$             ;NO, CONTINUE
1758  011150  012767  100001  170316          MOV     #100001,LGHOLD  ;YES, LOAD ERROR LOG ENABLE BIT
```

```
                                                                ;AND FREEZE ON FIRST ERROR BIT
011156                                          WRITE   RDWHMI,WTWHMI   ;ENABLE 11/60 ERROR LOG
011156  010046                                  MOV     R0,-(SP)        ;SAVE REG. 0
011160  076600  000022                          MED,RDWHMI              ;SAVE
011164  056700  170304                          BIS     LGHOLD,R0
011170  076600  000222                          MED,WTWHMI
011174  012600                                  MOV     (SP)+,R0        ;RESTORE REG. 0
011176  017767  170274  170274                  MOV     @CONTRL,KONTRL  ;GET CURRENT STATE
011204  052767  000200  170266                  BIS     #200,KONTRL     ;ADD ABORT BIT
011212  016777  170262  170256                  MOV     KONTRL,@CONTRL  ;WRITE IT BACK  ********************************
011220  112767  177777  167575  11$:    MOVB    #-1,WBFLG       ;INITIALIZE BUFFER ROTATION FLAG
011226  012702  026522                  MOV     #MODQ,R2        ;SETUP TO COUNT SELECTED MODULES, AND
                                                                ;IF NOT SYSERR OR POWER FAIL RESTART
                                                                ;CLEAR MODULES' PASCNT AND ERRCNT.
011232  012201                  1$:     MOV     (2)+,R1         ;MODULE ADDR TO R1.
011234  001462                          BEQ     RUNC            ;BP IF NO MORE.
011236  105767  170047                  TSTB    RMODE           ;INTERNAL RESTART?
011242  001012                          BNE     2$              ;BR IF YES.
011244  105767  170042                  TSTB    CHN             ;IN CHAIN MODE?
011250  100407                          BMI     2$              ;BR IF YES.
011252  005061  000026                  CLR     PSCNT(R1)       ;CLEAR MOD'S PASCNT.
011256  005061  000030                  CLR     ERCNT(R1)       ;CLEAR MOD'S ERROR COUNT.
011262  042761  020000  000020          BIC     #BIT13,STAT(R1) ;CLEAR STOPPED BIT.
011270  032761  040000  000020  2$:     BIT     #BIT14,STAT(R1) ;MODULE SELECTED?
011276  001755                          BEQ     1$              ;BR IF NOT.
011300  032761  020000  000020          BIT     #BIT13,STAT(R1) ;MODULE STOPPED?
011306  001351                          BNE     1$              ;BP IF YES.
011310  005767  167726                  TST     OFFSET          ;IF OFFSET 0, IGNORE BITS 10+11
011314  001414                          BEQ     3$
011316  032761  004000  000020          BIT     #BIT11,STAT(R1) ;NOT 0= IF BIT11 IS SET, DON'T COUNT IT
011324  001342                          BNE     1$
011326  032767  001777  167706          BIT     #1777,OFFSET    ;32K BOUNDARY?
011334  001404                          BEQ     3$              ;YES= IGNORE BIT 10
011336  032761  002000  000020          BIT     #BIT10,STAT(R1) ;NO= BIT 10 SET?
011344  001332                          BNE     1$              ;YES= DON'T RUN THIS MODULE
011346  105267  167443          3$:     INCB    MODCNT          ;NO, UP COUNT OF RUNABLE MODULES.
011352  005761  000020                  TST     STAT(R1)                ;BKMOD?
011356  100402                          BMI     .+6             ;NO= BRANCH
011360  105267  167441                  INCB    BKCNT           ;YES= UP COUNT
011364  032761  010000  000020          BIT     #BIT12,STAT(R1) ;EXTENDED MODULE?
011372  001402                          BEQ     .+6             ;NO= BRANCH
011374  105267  167426                  INCB    XCNT            ;YES= COUNT IT
011400  000714                          BR      1$              ;GO CHECK NEXT MODULE.
011402  105767  167407          RUNC:   TSTB    MODCNT          ;ANY RUNABLE MODULES?
011406  001005                          BNE     5$              ;YES, BRANCH
011410  005767  167626                  TST     OFFSET          ;NO, OFFSET 0?
011414  001501                          BEQ     RUND            ;YES= COMMAND INVALID
011416  000167  173572                  JMP     ENDSVB          ;NO= THE ONLY MODS RUNNABLE MUST BE THOSE
                                                                ;WITH BITS 10 OR 11 SET= RELOCATE
011422  116767  167367  167366  5$:     MOVB    MODCNT,MODCTR   ;LOAD COUNT OF RUNNABLE MODULES
                                                                ;INTO STARTUP COUNTER
011430  012767  026520  167402          MOV     #MODQ-2,MODPTR  ;MODULE TABLE ADDR TO MODPTR.
011436  012767  040000  167414          MOV     #40000,IOBKID   ;START WITH SPECIAL BACKGROUND MODULES.
011444  105267  167702                  INCB    NSTOP           ;SET THE NON-STOP FLAG
011450  105767  167637                  TSTB    ROTI
011454  001447                          BEQ     2$
```



```
011456  116767  167344  167345          MOVB    XCNT,ROTCT      ;SETUP ROTATION TIMEOUT COUNTER= 16 TIMES NUMBER
                                                                ;OF EXTENDED MODULES
011464  106367  167341                  ASLB    ROTCT
011470  106367  167335                  ASLB    ROTCT
011474  106367  167331                  ASLB    ROTCT
011500  106367  167325                  ASLB    ROTCT
011504  126727  167316  000020          CMPB    XCNT,#20        ;PREVENT LOSS OF COUNT IF OVERFLOW
011512  103403                          BLO     1$
011514  112767  000377  167307          MOVB    #377,ROTCT
011522  116767  167303  167604  1$:     MOVB    ROTCT,@GWBFC    ;INITIALIZE WRITE BUFFER ROTATION COUNTER
011530  116767  167261  167275          MOVB    MODCNT,PASREL   ;SET ALTERNATE RELOCATION COUNT TO 8
011536  106367  167271                  ASLB    PASREL          ;TIMES MODCNT
011542  106367  167265                  ASLB    PASREL
011546  106367  167261                  ASLB    PASREL
011552  126727  167237  000040          CMPB    MODCNT,#40
011560  103403                          BLO     3$
011562  112767  000377  167243          MOVB    #377,PASREL
011570  004767  000050          3$:     JSR     PC,SETWBF
011574  105767  167511          2$:     TSTB    RMODE
011600  001005                          BNE     4$
011602  104406  022737                  MSG$,DOT
011606  112767  000001  167475          MOVB    #1,RMODE        ;SET RMODE IN CURRENT BANK
011614  000167  170342          4$:     JMP INPUT
011620  005067  167420          RUND:   CLR     OFFSTD
011624  004767  007644                  JSR     PC,MVCODE
011630  105067  167455                  CLRB    RMODE
011634  104406  023316                  MSG$,INVCD4             ;'NO MODULES SELECTED'
011640  000167  177002                  JMP     COMCON
```

```
                                        ;SETUP WRITE BUFFER INFORMATION
                                        
011644  004567  003022          SETWBF: JSR     R5,TO18S
011650  001242                          OFFSET
011652  001232                          WBUF
011654  001234                          WBFEA
011656  066767  167336  167346          ADD     LOCORE,WBUF
011664  103003                          BCC     .+10
011666  062767  000020  167340          ADD     #20,WBFEA
011674  016767  167324  167334          MOV     LCOR12,WBUF12
011702  066767  167334  167326          ADD     OFFSET,WBUF12
011710  026767  167322  167310          CMP     WBUF12,HCOR12   ;IS WRITE BUFFER AT THE TOP ?
011716  001003                          BNE     SETWB1          ;NO, IT'S OK, CONTINUE
011720  004767  004726                  JSR     PC,ROTLOW       ;YES, SETUP TO ROTATE BELOW PROGRAM
011724  000207                          RTS     PC              ;RETURN
                                        
011726  016767  167274  167322  SETWB1: MOV     HCOR12,MXWBF    ;CALCULATE MAX SIZE OF WRITE BUFFER
011734  105767  167352                  TSTB    CHN             ;IN CHAIN MODE ?
011740  100003                          BPL     1$              ;NO, DON'T ADD ANY MORE WRITE BUFFER
011742  062767  000040  167306          ADD     #40,MXWBF       ;YES, ADD 1K TO MAX. WRITE BUFFER SIZE
011750  166767  167262  167300  1$:     SUB     WBUF12,MXWBF
011756  006367  167274                  ASL     MXWBF
011762  006367  167270                  ASL     MXWBF
011766  006367  167264                  ASL     MXWBF
011772  006367  167260                  ASL     MXWBF
011776  006367  167254                  ASL     MXWBF
012002  103003                          BCC     .+10
012004  012767  177777  167244          MOV     #177777,MXWBF
012012  000207                          RTS     PC
```



```
                                        ;KT11 INITIALIZATION ROUTINE ASSUMES KT11 IS OFF WHEN CALLED
                                        
012014  012737  015254  000250  KTINIT: MOV     #KTERR,@#250    ;SETUP MEMORY MANAGEMENT ABORT VECTOR
012022  012737  000340  000252          MOV     #PRTY7,@#252    ;DON'T LET ANYTHING ABORT A MEMORY MANAGEMENT ABORT
012030  012701  001276                  MOV     #PDPTAB,R1      ;SETUP TO CLEAR ALL KT11 REGISTERS
012034  012737  000006  000004          MOV     #6,@#4          ;SETUP TO RTI IF KT11-D WHEN
012042  012737  000002  000006          MOV     #RTI,@#6        ;ADDRESSING REGISTERS IT LACKS
012050  012703  000040          1$:     MOV     #32.,R3         ;LOAD THE COUNT OF KT11 REGISTERS TO BE CLEARED
012054  012102                          MOV     (R1)+,R2        ;GET ADDRESS OF REGISTER SET
012056  005012                  2$:     CLR     (R2)            ;CLEAR ALL PAR'S AND PDR'S
012060  062702  000002                  ADD     #2,R2           ;USE THE ADD INSTRUCTION TO POINT TO THE NEXT REGISTER.
012064  005303                          DEC     R3              ;REDUCE COUNT OF APR REGISTERS CLEARED
012066  001373                          BNE     2$              ;IS COUNT EXCEEDED? IF NO, GO CLEAR AGAIN
012070  020127  001302                  CMP     R1,#PDREND      ;IF YES, THEN IS THE TABLE BOUNDARY EXCEEDED?
012074  003765                          BLE     1$              ;IF NO, DO THE NEXT SET IN THE TABLE
012076  012737  015246  000004          MOV     #BUSERR,@#4     ;RESTORE THE BUS ERROR VECTOR.
012104  012737  000340  000006          MOV     #PRTY7,@#6      ;RESTORE THE BUS ERROR PRIORITY
012112  012737  140000  177776          MOV     #UMODE, @#PS    ;SET TO USER MODE
012120  012706  027002                  MOV     #USP,SP         ;LOAD USER STACK POINTER
012124  105767  167203                  TSTB    MDL45           ;IS THIS A PDP 11/45?
012130  001405                          BEQ     3$              ;IF NOT, DO NOT SET UP THE SUPERVISOR STACK
012132  012737  040000  177776          MOV     #SMODE,@#PS     ;IF YES, SET TO SUPERVISOR MODE
012140  012706  027012                  MOV     #SSP,SP         ;LOAD SUPERVISOR STACK POINTER
012144  005037  177776          3$:     CLR     @#PS            ;RETURN TO KERNEL MODE
                                        
                                        ;SET UP ALL THE PAGE DESCRIPTIONS THE SAME: READ/WRITE PROTECTION
                                        ;UPWARD PAGE ADDRESSING,AND MAXIMUM BLOCK COUNT.
                                        ; (177 OCTAL BLOCKS = 128 X 32 WORD DECIMAL BLOCKS)
                                        
012150  004467  002564          KTSET0: JSR     R4,SAV04        ;SAVE REGS. 0 THRU 4
012154  005001                          CLR     R1              ;MAP STARTING AT BANK 0
012156  012702  172300                  MOV     #KIPDR0,R2      ;LOAD ADDRESS OF PDR 0
012162  012703  000007                  MOV     #7,R3           ;LOAD COUNTER TO DO 28K
012166  012712  077406          1$:     MOV     #77406,(R2)     ;SET THE PDR TO 4K, R/W, UP
012172  010162  000040                  MOV     R1,40(R2)       ;SET THE PAR TO THE PROPER 4K BANK
012176  005722                          TST     (R2)+           ;POINT TO NEXT PDR AND PAR
012200  062701  000200                  ADD     #200,R1         ;STEP TO NEXT 4K BANK
012204  005303                          DEC     R3              ;ALL DONE ?
012206  001367                          BNE     1$              ;NO, CONTINUE
                                        
012210  012737  007600  172356          MOV     #7600,@#KIPAR7  ;POINT REGISTER 7 THE UNIBUS ADDRESS SPACE
012216  012737  077406  172316          MOV     #77406,@#KIPDR7 ;SETS UP REGISTER 7
012224  004767  002522                  JSR     PC,RST04        ;RESTORE REGS. 0 THRU 4
012230  000207                          RTS     PC              ;RETURN TO THE CALLING PROCEDURE
```

```
 1916                                   ;PARITY ENABLE ROUTINE --- ASSUMES THAT PARITY IS PRESENT
 1917
 1918  012232  132767  000002  167100  PAENBL: BITB    #BIT1,PPRES     ;IS IT AN 11/60 ?    ****************************
 1919  012240  001403                          BEQ     1$              ;NO, CONTINUE
 1920  012242  052777  000001  167226          BIS     #1,@CONTRL      ;ENABLE CACHE PARITY TRAPS  ********************
 1921
 1922  012250  012700  001430          1$:     MOV     #PARTAB,R0      ;GET ADDRESS OF PARITY ADDRESSES
 1923  012254  005001                          CLR     R1              ;CLEAR COUNTER
 1924
 1925  012256  012770  000001  000000  2$:     MOV     #1,@(R0)        ;SET PARITY ENABLE
 1926  012264  032770  000001  000000          BIT     #1,@(R0)        ;IS PARITY ENABLED ?
 1927  012272  001002                          BNE     3$              ;YES, CONTINUE
 1928  012274  004767  003476                  JSR     PC,PAMSG        ;NO, REPORT THE ERROR
 1929
 1930  012300  005720                  3$:     TST     (R0)+           ;STEP TO NEXT REGISTER ADDRESS
 1931  012302  005201                          INC     R1              ;STEP THE REGISTER COUNTER
 1932  012304  126701  167031                  CMPB    CNT1,R1         ;ALL PARITY REGISTERS SET ?
 1933  012310  003362                          BGT     2$              ;NO, DO THE NEXT ONE
 1934
 1935  012312  000207                  4$:     RTS     PC              ;YES, ALL DONE, RETURN
 1936
```



```
 1937                                   ;MAP ROUTINE, TYPES RESIDENT MODULES AND THEIR START ADDRESS.
 1938
 1939  012314  105267  166471  MAP:    INCB    DIRIND          ;SET DIR INDICATOR.
 1940  012320  105067  013303          CLRB    MDIRE           ;TERMINATE ASCII STRING EARLY.
 1941  012324  004767  000004          JSR     PC,DIRA         ;TYPE MAP.
 1942  012330  000167  176312          JMP     COMCON
 1943
 1944  012334  012702  026522  DIRA:   MOV     #MODQ,R2        ;GET MODULE TABLE ADDR.
 1945  012340  012201          1$:     MOV     (2)+,R1         ;GET MODULE ADDR.
 1946  012342  001446                  BEQ     5$              ;BR IF 0, ALL DONE.
 1947  012344  032761  040000  000020  BIT     #BIT14,STAT(R1) ;MODULE SELECTED?
 1948  012352  001003                  BNE     2$              ;BR IF YES.
 1949  012354  105767  166431          TSTB    DIRIND          ;TYPING DIRECTORY?
 1950  012360  001767                  BEQ     1$              ;BR IF NOT, DONT TYPE UNSELECTED MODS.
 1951  012362  004567  001246  2$:     JSR     R5,FILLNM       ;FILL MOD NAME IN ASCII STRING.
 1952  012366  025574                  AMODNM+1                ;ADDR TO STUFF NAME IN.
 1953  012370  010146                  MOV     R1,-(SP)                ;MODULE ADDR TO STACK.
 1954  012372  004567  001542          JSR     R5,OACNV        ;CONVERT MOD ADDR TO ASCII.
 1955  012376  025605                  APC
 1956  012400  016146  000020          MOV     STAT(R1),-(SP)  ;CONVERT MODULE STATUS.
 1957  012404  004567  001530          JSR     R5,OACNV
 1958  012410  025621                  AMDSTA
 1959  012412  016146  000026          MOV     PSCNT(R1),-(SP) ;MOD'S PASS COUNT TO STACK.
 1960  012416  004567  001750          JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
 1961  012422  025640                  APSCNT
 1962  012424  016146  000030          MOV     ERCNT(R1),-(SP) ;MOD'S ERROR COUNT TO STACK.
 1963  012430  004567  001736          JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
 1964  012434  025657                  AERRS
 1965  012436  105767  166347          TSTB    DIRIND          ;TYPING DIRECTORY?
 1966  012442  001003                  BNE     3$              ;BR IF YES.
 1967  012444  112767  000040  013155  MOVB    #40,MDIRE       ;NO, ALLOW TYPING FULL STRING.
 1968  012452  104406  025573  3$:     MSG$,AMODNM             ;TYPE ASCII LINE.
 1969  012456  000730                  BR      1$              ;DO IT AGAIN.
 1970  012460  000207          5$:     RTS     PC              ;EXIT.
```

```
                                        ;SELECT MODULE(S) ROUTINE.
                                        ;
012462  012767  052761  000040  SEL:    MOV     #52761,SLDSB    ;MODIFY COMMON TO SELECT MODULE(S).
012470  012767  052761  000054          MOV     #52761,SLDSE    ;
012476  000406                          BR      SLDS            ;GO TO COMMON.
                                        ;
                                        ;DESELECT MODULE(S) ROUTINE.
                                        ;
012500  012767  042761  000022  DES:    MOV     #42761,SLDSB    ;MODIFY COMMON TO DESELECT MODULE(S).
012506  012767  042761  000036          MOV     #42761,SLDSE
                                        ;
                                        ;SELECT/DESELECT COMMON.
                                        ;
012514                                  SLDS:
012514  004767  001142          SLDSA:  JSR     PC,GETNAM       ;CHECK NAME, GET MODULE ADDR.
012520  000410                          BR      SLDSC           ;NO NAME, SELECT/DESELECT ALL.
012522  000405                          BR      SLDSL           ;INVALID OR NEX NAME.
012524  016701  166310                  MOV     MODPTR,R1       ;MODULE ADDR TO R1.
012530  042761  040000  000020  SLDSB:  BIC     #BIT14,STAT(R1) ;SELECT/DESELECT MODULE.
012536  000167  176104          SLDSL:  JMP     COMCON          ;DONE.
012542  012702  026522          SLDSC:  MOV     #MODQ,R2        ;MODULE TABLE ADDR TO R2.
012546  012201                  SLDSD:  MOV     (2)+,R1         ;GET MODULE ADDR.
012550  001772                          BEQ     SLDSL           ;BR IF 0,END OF TABLE, DONE.
012552  042761  040000  000020  SLDSE:  BIC     #BIT14,STAT(R1) ;SELECT/DESELECT MODULE.
012560  000772                          BR      SLDSD           ;DO IT AGAIN.
                                        ;LOCATIONS SLDSB AND SLDSE ARE PURE WHILE IN RUN MODE.
```



```
                                        ;KT11 DISABLE ROUTINE
                                        ;
012562  105767  166516          KTOFF:  TSTB    KTPRES
012566  001002                          BNE     1$
012570  000167  176062                  JMP     COMCO3          ;IGNORE COMMAND- NO KT OR ALREADY OFF
012574  105067  166504          1$:     CLRB    KTPRES
                                        ;
                                        ;
012600  005037  177572                  CLR     @#SSR0          ;TURN OFF KT11
012604  026727  166416  001600          CMP     HCOR12,#1600    ;MORE THAN 28K AVAILABLE ?
012612  101403                          BLOS    2$              ;NO, CONTINUE
012614  012767  001600  166404          MOV     #1600,HCOR12    ;YES, SET MEMORY SIZE TO 28K
012622  104406  024007          2$:     MSG$,KTDIS              ;"KT11 OFF"
012626  000167  176024                  JMP     COMCO3
                                        ;
                                        ;KT11 ENABLE ROUTINE
                                        ;
012632  012737  012704  000004  KTON:   MOV     #1$,@#4
012640  032737  000001  177572          BIT     #1,@#SSR0       ;ALREADY ON ?
012646  001027                          BNE     2$              ;YES, FORGET IT
012650  112767  000001  166426          MOVB    #1,KTPRES       ;NO, SET KT PRESENT FLAG
012656  016767  166346  166342          MOV     HCORSV,HCOR12   ;RESTORE MEMORY SIZE
012664  004767  177124                  JSR     PC,KTINIT       ;INITIALIZE KT11
012670  012737  000001  177572          MOV     #1,@#SSR0       ;TURN IT ON
                                        ;
                                        ;
012676  104406  024615                  MSG$,KTONM              ;"KT11 ENABLED"
012702  000411                          BR      2$              ;CONTINUE
012704  012737  015246  000004  1$:     MOV     #BUSERR,@#4
012712  005067  165060                  CLR     PSW
012716  104406  023776                  MSG$,NOKT               ;"NO KT11"
012722  000167  175720                  JMP     COMCON          ;CONTINUE
012726  012737  015246  000004  2$:     MOV     #BUSERR,@#4
012734  005067  165036                  CLR     PSW
012740  000167  175712                  JMP     COMCO3
```

```
                                        ;THIS ROUTINE ENABLES WRITE BUFFER ROTATION
                                        ;
012744  105067  166344          ROTON:  CLRB    ROTL            ;CLEAR ROTATION LOCK FLAG
012750  004767  003764                  JSR     PC,ROTCHK       ;GO DO "ENABLED" MESSAGE
012754  000167  175676                  JMP     COMCO3          ;CONTINUE
                                        ;
                                        ;
                                        ;THIS ROUTINE DISABLES WRITE BUFFER ROTATION
                                        ;
012760  112767  000377  166326  ROTOFF: MOVB    #377,ROTL       ;SET THE ROTATION LOCK FLAG
012766  016767  166326  166322          MOV     ROTNUM,ROTCNT   ;LOAD THE ROTATION COUNTER
012774  104406  024555                  MSG$,ROTDIS             ;"WBUF ROTATION DISABLED"
013000  000167  175652                  JMP     COMCO3          ;CONTINUE
```



```
                                        ;THIS ROUTINE IS NECESSARY TO SUPPORT THE PON COMMAND
                                        ;
013004  004767  000004          PON:    JSR     PC,PONN         ;GO TURN ON PARITY
013010  000167  175642                  JMP     COMCO3          ;CONTINUE
                                        ;
                                        ;
                                        ;THIS ROUTINE WILL TURN ON PARITY IF AVAILABLE
                                        ;
013014  132767  000001  166316  PONN:   BITB    #BIT0,PPRES             ;ANY PARITY TO TURN ON ?
013022  001405                          BEQ     1$              ;NO, SAY SO AND RETURN
013024  004767  177202                  JSR     PC,PAENBL       ;YES, GO TURN IT ON
013030  104406  023756                  MSG$,PENABL             ;"PARITY ENABLED"
013034  000402                          BR      2$              ;RETURN
013036  104406  024021          1$:     MSG$,NOPAR              ;"NO PARITY"
013042  000207                  2$:     RTS     PC              ;RETURN
                                        ;
                                        ;
                                        ;THIS ROUTINE DISABLES PARITY IF PRESENT
                                        ;
013044  132767  000001  166266  POFF:   BITB    #BIT0,PPRES             ;ANY PARITY TO TURN OFF ?
013052  001003                          BNE     1$              ;YES, CONTINUE
013054  104406  024021                  MSG$,NOPAR              ;"NO PARITY"
013060  000421                          BR      4$              ;RETURN
013062  132767  000002  166250  1$:     BITB    #BIT1,PPRES     ;IS IT AN 11/60 ?     ****************************
013070  001403                          BEQ     2$              ;NO, CONTINUE
013072  042777  000001  166376          BIC     #1,@CONTRL      ;DISABLE CACHE PARITY TRAPS **************
                                        ;
013100  012700  001430          2$:     MOV     #PARTAB,R0      ;GET ADDRESS OF PARITY ADDRESSES
013104  005001                          CLR     R1              ;CLEAR THE COUNTER
                                        ;
013106  005030                  3$:     CLR     @(R0)+          ;CLEAR THE PARITY REGISTER
013110  005201                          INC     R1              ;STEP THE REGISTER COUNTER
013112  126701  166223                  CMPB    CNT1,R1         ;ALL PARITY REGISTERS CLEARED ?
013116  003373                          BGT     3$              ;NO, CLEAR THE NEXT ONE
                                        ;
013120  104406  024034                  MSG$,PAOFF              ;"PARITY OFF"
013124  000167  175526          4$:     JMP     COMCO3          ;CONTINUE
                                        ;
                                        ;THIS ROUTINE TURNS ON THE CACHE, SUPPORTS THE CON COMMAND.
                                        ;
013130  010046                  CON:    MOV     R0,-(SP)        ;SAVE R0
013132  012700  000200                  MOV     #200,R0         ;SWEEP CACHE CLEAN FIRST
013136  076600                          MED
013140  000352                          352
013142  012600                          MOV     (SP)+,R0        ;RESTORE R0
013144  042767  000014  166326          BIC     #14,KONTRL      ;CLEAR OUT THE FORCE MISS BITS
013152  016777  166322  166316          MOV     KONTRL,@CONTRL  ;DO IT
013160  104406  023643                  MSG$,CONN               ;"CACHE ENABLED"
013164  000167  175466                  JMP     COMCO3          ;CONTINUE
                                        ;
                                        ;
                                        ;THIS ROUTINE TURNS OFF THE CACHE, SUPPORTS THE COFF COMMAND.
                                        ;
013170  052767  000014  166302  COFF:   BIS     #14,KONTRL      ;SET THE FORCE MISS BITS
013176  016777  166276  166272          MOV     KONTRL,@CONTRL  ;DO IT
```

```
 2101  013204  104406  023662                  MSG6,COFFF              ;"CACHE DISABLED"
 2102  013210  000167  175442                  JMP     COMCO3
 2103
```



```
 2104                                  ;MODIFY ROUTINE.
 2105                                  ;GETS ADDRESS AND PRINTS ADDR CONTENTS.
 2106                                  ;USER CAN INPUT NEW NUMBER OR CR.
 2107                                  ;CR MEANS HE JUST WANTS TO EXAMINE.
 2108
 2109  013214  112767  177777  166102  FILL:   MOVB    #-1,FILLID      ;SET FILL INDICATOR.
 2110  013222                          MOD:
 2111  013222  004767  000434          5$:     JSR     PC,GETNAM       ;GET NAME AND MODULE ADDR.
 2112  013226  000401                          BR      4$              ;NO NAME.
 2113  013230  000440                          BR      3$              ;INVALID OR NEX NAME.
 2114  013232  004767  000212          4$:     JSR     PC,GETADR       ;GET ADDRESS.
 2115  013236  000430                          BR      6$              ;INVALID ADDR OR NO ADDR.
 2116  013240  016704  165576                  MOV     ADDRLO,R4               ;MOVE ADDR TO R4.
 2117  013244  066704  165570                  ADD     MODPTR,R4       ;ADD MODULE ADDR TO ADDR.
 2118  013250  010446                  1$:     MOV     R4,-(SP)                ;TYPE ADDR.
 2119  013252  004767  000150                  JSR     PC,ITOA         ;
 2120  013256  011446                  7$:     MOV     (R4),-(SP)      ;GET CONTENTS OF ADDR.
 2121  013260  004767  000142                  JSR     PC,ITOA         ;PRINT IT OUT.
 2122  013264  004767  175376                  JSR     PC,SINPUT       ;GET NEW CONTENTS.
 2123  013270  004767  000222                  JSR     PC,GETNUM       ;CONVERT INPUT TO BINARY.
 2124  013274  000416                          BR      3$              ;INVALID DATA (NOT A NUMBER).
 2125  013276  005767  165560                  TST     YES             ;NUMBER?
 2126  013302  001401                          BEQ     2$              ;BR IF NOT.
 2127  013304  010114                          MOV     R1,(R4)         ;STORE NEW VALUE.
 2128  013306  122702  000012          2$:     CMPB    #12,R2          ;LINE FEED?
 2129  013312  001007                          BNE     3$              ;NO, DONE.
 2130  013314  005724                          TST     (R4)+           ;POINT TO NEXT LOCATION.
 2131  013316  000754                          BR      1$              ;REPEAT FOR NEXT LOC.
 2132  013320  012704  001306          6$:     MOV     #FILCNT,R4      ;GET FILCNT ADDR.
 2133  013324  105767  165774                  TSTB    FILLID          ;IN FILL MODE?
 2134  013330  001352                          BNE     7$              ;BR IF YES.
 2135  013332  000167  175310          3$:     JMP     COMCON          ;DONE.
```

```
 2136                                   ;BINARY TO ASCII TYPE ROUTINE, TYPES NUMBER(S) ON STACK.
 2137
 2138  013336  016646  000002          ITOAX:  MOV     2(SP),-(SP)     ;LOAD EA BITS
 2139  013342  016666  000002  000004          MOV     2(SP),4(SP)
 2140  013350  016666  000006  000002          MOV     6(SP),2(SP)
 2141  013356  004567  000570                  JSR     R5,OACNVX
 2142  013362  025675                          HOLD6                   ;AND PUT IT HERE
 2143  013364  104406  025675                  MSG$,HOLD6              ;TYPE OUT THE VALUE
 2144  013370  000425                          BR      ITOUT
 2145
 2146  013372  016646  000002          ITOA8:  MOV     2(SP),-(SP)     ;LOAD UPPER BITS
 2147  013376  016666  000002  000004          MOV     2(SP),4(SP)     ;AND LOWER BITS
 2148  013404  016666  000006  000002          MOV     6(SP),2(SP)     ;AND ADJUST THE STACK
```



```
 2149  013412  004567  000542                  JSR     R5,OACNV8       ;CONVERT 8 OCTAL NUMBERS TO ASCII
 2150  013416  025706                          HOLD8                   ;AND PUT THEM HERE
 2151  013420  104406  025706                  MSG$,HOLD8              ;TYPE OUT THE OCTAL VALUE
 2152  013424  000407                          BR      ITOUT           ;GO TYPE IT OUT
 2153
 2154
 2155  013426  016646  000002          ITOA:   MOV     2(SP),-(SP)
 2156  013432  004567  000502                  JSR     R5,OACNV        ;CONVERT NUMBER TO ASCII,
 2157  013436  025675                          HOLD6
 2158  013440  104406  025675                  MSG$,HOLD6              ;TYPE OCTAL VALUE,
 2159  013444  012616                  ITOUT:  MOV     (SP)+,(SP)
 2160  013446  000207                          RTS     PC              ;EXIT,
```

```
                                ;THIS ROUTINE GETS THE ADDRESS ASSOCIATED WITH THE MOD COMMAND
                                ;
013450  004767  000042          GETADR: JSR     PC,GETNUM       ;GET NUMBER
013454  000414                          BR      1$              ;INVALID DATA RETURN.
013456  005767  165400                  TST     YES             ;NUMBER?
013462  001411                          BEQ     1$              ;BR IF NOT.
013464  006201                          ASR     R1              ;GOOD ADDRESS?
013466  103410                          BCS     ADRERR          ;NOGOOD, IT'S ODD
013470  060101                          ADD     R1,R1           ;RESTORE THE ADR
013472  010167  165344                  MOV     R1,ADDRLO       ;STORE AT ADDRLO
013476  010367  165342                  MOV     R3,ADDRHI       ;AND ANY HIGH ORDER BITS IN ADDRHI
013502  062716  000002                  ADD     #2,(SP)         ;SET UP GOOD EXIT.
013506  000207                  1$:     RTS     PC              ;DONE
                                ;
013510  104406  023344          ADRERR: MSG$,INVADR             ;TYPE INVALID ADDR OR DATA.
013514  000207                          RTS     PC              ;EXIT.
```



```
                                ;GET NUMBER FROM KEYBOARD BUFFER, AND RETURN THE LOW ORDER BITS
                                ;IN R1 AND THE HIGH ORDER BITS IN R3.
                                ;ROUTINE ADDS 2 TO THE RETURN PC IF SUCCESSFUL.
                                ;
013516  016700  165314          GETNUM: MOV     KBPTR,R0        ;GET BUFFER POINTER.
013522  005001                  3$:     CLR     R1              ;DATA
013524  005003                          CLR     R3
013526  005067  165330                  CLR     YES             ;FLAG
013532  112002                  1$:     MOVB    (R0)+,R2        ;GET A BYTE
013534  120227  000040                  CMPB    R2,#40          ;SPACE?
013540  001770                          BEQ     3$              ;YES,IGNORE IT
013542  122702  000015                  CMPB    #15,R2          ;CR?
013546  001425                          BEQ     2$              ;YES,RETURN
013550  120227  000012                  CMPB    R2,#12          ;LF?
013554  001422                          BEQ     2$              ;DONE
013556  120227  000060                  CMPB    R2,#'0          ;LOW LIMIT
013562  002752                          BLT     ADRERR          ;TOO LOW, REPORT THE ERROR
013564  120227  000067                  CMPB    R2,#'7          ;HIGH LIMIT
013570  003347                          BGT     ADRERR          ;TOO HIGH, REPORT THE ERROR
013572  006301                          ASL     R1              ;SHIFT OLD STUFF
013574  006103                          ROL     R3
013576  006301                          ASL     R1              ;3 TIMES LEFT
013600  006103                          ROL     R3
013602  006301                          ASL     R1              ;I.E. MULT. BY OCTAL 10
013604  006103                          ROL     R3
013606  060201                          ADD     R2,R1           ;ADD NEW TO OLD
013610  162701  000060                  SUB     #'0,R1          ;BUT GET RID OF ASCII STUFF
013614  005267  165242                  INC     YES             ;SET FLAG
013620  000744                          BR      1$              ;MORE, MORE
013622  010067  165210          2$:     MOV     R0,KBPTR        ;SAVE CURRENT BUF POINTER.
013626  062716  000002                  ADD     #2,(SP)         ;SET UP OK EXIT.
013632  000207                          RTS     PC              ;EXIT.
```

```
 2209                                   ;THIS ROUTINE PUTS THE MODULE NAME INTO AN ASCII STRING FOR OUTPUT
 2210
 2211  013634  010346                   FILLNM: MOV     R3,-(SP)                ;SAVE R3.
 2212  013636  012503                           MOV     (R5)+,R3                ;STORE ADDR TO R3.
 2213  013640  012704  000005                   MOV     #5,R4           ;WILL DO 5 TIMES.
 2214  013644  112123                   1$:     MOVB    (R1)+,(R3)+     ;MOVE CHAR.
 2215  013646  005304                           DEC     R4              ;DONE?
 2216  013650  001375                           BNE     1$              ;BR IF NOT.
 2217  013652  162701  000005                   SUB     #5,R1           ;RESTORE R1.
 2218  013656  012603                           MOV     (SP)+,R3                ;RESTORE R3.
 2219  013660  000205                           RTS     R5              ;EXIT.
```



```
 2220                                   ;THIS ROUTINE GETS THE MODULE NAME ASSOCIATED WITH THE MOD AND SEL/DES COMMANDS.
 2221
 2222  013662  016700  165150           GETNAM: MOV     KBPTR,R0        ;GET BUFFER POINTER.
 2223  013666  005067  165170                   CLR     YES             ;CLEAR SUCCESS INDICATOR.
 2224  013672  005067  165142                   CLR     MODPTR          ;IF 0, ABS ADDR.
 2225  013676  012702  000005                   MOV     #5,R2           ;WILL DO 5 TIMES.
 2226  013702  012701  025574                   MOV     #AMODNM+1,R1    ;WILL STORE KYBD NAME IN TMPNAM.
 2227  013706  112721  000040           1$:     MOVB    #40,(R1)+       ;BUT CLEAR IT TO SPACES FIRST.
 2228  013712  005302                           DEC     R2              ;DONE?
 2229  013714  001374                           BNE     1$              ;BR IF NOT.
 2230  013716  012701  025574                   MOV     #AMODNM+1,R1    ;RELOAD TEMP ADDR.
 2231  013722  111002                   2$:     MOVB    (R0),R2         ;GET CHAR FROM KBUF AREA.
 2232  013724  122702  000015                   CMPB    #15,R2          ;CR?
 2233  013730  001442                           BEQ     6$              ;BR IF YES.
 2234  013732  122702  000012                   CMPB    #12,R2          ;LF? TREAT AS CR.
 2235  013736  001437                           BEQ     6$              ;BR IF YES.
 2236  013740  122702  000040                   CMPB    #40,R2          ;SPACE? DISREGARD LEADING ONES.
 2237  013744  001430                           BEQ     5$              ;BR IF YES.
 2238  013746  120227  000101                   CMPB    R2,#'A          ;A OR HIGHER?
 2239  013752  002403                           BLT     3$              ;BR IF LESS. (COULD BE A NUMBER).
 2240  013754  120227  000132                   CMPB    R2,#'Z          ;Z OR LESS?
 2241  013760  003411                           BLE     4$              ;BR IF Z OR LESS.
 2242  013762  120227  000060           3$:     CMPB    R2,#'0          ;0 OR HIGHER?
 2243  013766  002427                           BLT     7$              ;BR IF LESS. YOU LOOSE!
 2244  013770  120227  000071                   CMPB    R2,#'9          ;9 OR LESS?
 2245  013774  003024                           BGT     7$              ;BR IF GREATER. YOU LOOSE!.
 2246  013776  005767  165060                   TST     YES             ;ANY PREVIOUS CHARS?
 2247  014002  001426                           BEQ     8$              ;BR IF NOT. NOT LEADING NUMERICS ALLOWED!.
 2248  014004  026727  165052  000005   4$:     CMP     YES,#5          ;6TH CHARACTER?
 2249  014012  002015                           BGE     7$              ;IF YES, BRANCH
 2250  014014  110221                           MOVB    R2,(R1)+                ;STORE CHAR.
 2251  014016  005267  165040                   INC     YES
 2252  014022  005200                   41$:    INC     R0              ;POINT TO NEXT CHAR.
 2253  014024  000736                           BR      2$              ;DO IT AGAIN!
 2254  014026  005767  165030           5$:     TST     YES             ;ANY CHARS?
 2255
 2256
 2257
 2258  014032  001773                           BEQ     41$             ;BR IF NOT.
 2259  014034  000416                           BP      9$
 2260
 2261  014036  005767  165020           6$:     TST     YES             ;ANY CHARS?
 2262  014042  001013                           BNE     9$              ;BR IF YES.
 2263  014044  000405                           BR      8$              ;NO NAME.
 2264  014046  104406  023370           7$:     MSG$,INVNAM             ;TYPE INVALID OR NEX (NON-EXISTENT) MESSAGE.
 2265  014052  062716  000002                   ADD     #2,(SP)         ;SET UP INVALID NAME EXIT.
 2266  014056  000207                           RTS     PC              ;EXIT.
 2267
 2268
 2269  014060  005067  164776           8$:     CLR     YES             ;
 2270  014064  010067  164746                   MOV     R0,KBPTR        ;SAVE CURRENT BUF POINTER.
 2271  014070  000207                           RTS     PC              ;EXIT.
 2272  014072  012702  026522           9$:     MOV     #MODQ,R2        ;MODULE TABLE ADDR TO R2.
 2273  014076  012203                   10$:    MOV     (R2)+,R3                ;GET MODULE ADDR.
 2274  014100  001762                           BEQ     7$              ;INVALID NAME, NO MODULE!!
 2275  014102  012701  025574                   MOV     #AMODNM+1,R1
```

```
 2276  014106  012704  000005            MOV     #5,R4           ;WILL MATCH 5 CHARS.
 2277  014112  122123                 11$: CMPB    (R1)+,(3)+      ;CHAR MATCH?
 2278  014114  001370                      BNE     10$             ;BR IF NOT.
 2279  014116  005304                      DEC     R4              ;DONE?
 2280  014120  001374                      BNE     11$             ;BR IF NOT.
 2281  014122  014267  164712            MOV     -(R2),MODPTR    ;MODULE ADDR TO MODPTR.
 2282  014126  010067  164704            MOV     R0,KBPTR        ;SAVE CURRENT BUF POINTER.
 2283  014132  062716  000004            ADD     #4,(SP)         ;SET UP SUCCESS EXIT.
 2284  014136  000207                      RTS     PC              ;EXIT.
```



```
 2285                                ;OCTAL TO ASCII CONVERT ROUTINE.
 2286                                ;CONVERTS ARGUMENT ON STACK TO ASCII, STORES STRING STARTING AT ADDRESS
 2287                                ;GIVEN FOLLOWING THE CALL
 2288                                ;EXTENDED ENTRY REQUIRES ADDITIONAL  ARGUMENT ON STACK- AFTER
 2289                                ;PUSHING LOWER 16 BITS ONTO STACK, PUSH EA BITS (SHIFTED TO POS 4,5)
 2290
 2291  014140  005067  164746    OACNV:  CLR     OASCEA          ;CLEAR OUT UPPER BITS -- NOT USED
 2292  014144  105067  165172            CLRB    ATE             ;CLEAR 8 CHARACTER FLAG
 2293  014150  000412                      BR      CONVRT          ;CONTINUE
 2294
 2295  014152  105067  165164    OACNVX: CLRB    ATE             ;CLEAR 8 CHARACTER FLAG
 2296  014156  000403                      BR      HOP             ;CONTINUE
 2297
 2298  014160  112767  000200  165154  OACNV8: MOVB    #BIT7,ATE       ;SET 8 CHARACTER FLAG
 2299
 2300  014166  016667  000002  164716  HOP:    MOV     2(SP),OASCEA
 2301  014174  012616                      MOV     (SP)+,(SP)
 2302  014176  004467  000536            CONVRT: JSR     R4,SAV04        ;SAVE REGS 0-4.
 2303  014202  016600  000014                    MOV     12.(SP),R0      ;GET OCTAL VALUE.
 2304  014206  012501                      MOV     (R5)+,R1                ;GET DEST ADDR.
 2305  014210  012702  000006            MOV     #6,R2           ;SET CONVERT COUNT TO 6.
 2306  014214  060201                      ADD     R2,R1           ;DEVELOP ADDR TO STORE 1ST CHAR.
 2307  014216  105767  165120            TSTB    ATE             ;SET UP TO DO 8 CHARACTERS ?
 2308  014222  100002                      BPL     3$              ;NO, CONTINUE
 2309  014224  062701  000002            ADD     #2,R1           ;YES, ADD 2 TO THE ADDRESS
 2310  014230  010003                 3$:  MOV     R0,R3           ;GET VALUE TO R3.
 2311  014232  042703  177770            BIC     #177770,R3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
 2312  014236  062703  000060            ADD     #60,R3          ;CONVERT TO ASCII.
 2313  014242  110341                      MOVB    R3,-(R1)                ;STORE IT.
 2314  014244  042700  000007            BIC     #7,R0           ;CLEAR DIGIT JUST CONVERTED.
 2315  014250  006000                      ROR     R0              ;SHIFT IN NEXT DIGIT.
 2316  014252  006000                      ROR     R0
 2317  014254  006000                      ROR     R0
 2318  014256  005302                      DEC     R2              ;DONE 6 DIGITS?
 2319  014260  001363                      BNE     3$              ;BR IF NOT.
 2320  014262  105767  165054            TSTB    ATE             ;SET UP TO DO 8 CHARACTERS ?
 2321  014266  100410                      BMI     4$              ;YES, DO IT
 2322  014270  016702  164616            MOV     OASCEA,R2
 2323  014274  006202                      ASR     R2
 2324  014276  006202                      ASR     R2
 2325  014300  006202                      ASR     R2
 2326  014302  060203                      ADD     R2,R3
 2327  014304  110311                      MOVB    R3,(R1)
 2328  014306  000425                      BR      5$              ;GO FINISH UP
 2329
 2330  014310  132767  000001  165024  4$:  BITB    #BIT0,ATE       ;ALL DONE ?
 2331  014316  001021                      BNE     5$              ;YES, GET OUT
 2332  014320  016700  164566            MOV     OASCEA,R0       ;NO, LOAD UPPER NUMBERS TO BE CONVERTED
 2333  014324  006300                      ASL     R0              ;LINE UP BITS BELONGING TO LOWER NUMBER
 2334  014326  042700  177770            BIC     #177770,R0      ;GIT RID OF ALL OTHERS
 2335  014332  060003                      ADD     R0,R3           ;ADD IT IN TO COMPLETE THE NUMBER
 2336  014334  110311                      MOVB    R3,(R1)         ;STORE IT
 2337  014336  016700  164550            MOV     OASCEA,R0       ;LOAD UPPER BITS AGAIN
 2338  014342  006200                      ASR     R0              ;GIT RID OF THE 2 BITS
 2339  014344  006200                      ASR     R0              ;JUST TAKEN CARE OF
 2340  014346  012702  000002            MOV     #2,R2           ;SET UP TO DO THE LAST 2 CHARACTERS
```

```
 2341  014352  152767  000001  164762          BISB    #BIT0,ATE       ;SET 'ALL DONE' FLAG
 2342  014360  000723                          BR      3$              ;GO DO THE LAST 2 CHARACTERS
 2343
 2344  014362  004767  000364          5$:     JSR     PC,RST04        ;RESTORE REGS, 0 THRU 4
 2345  014366  012616                          MOV     (SP)+,(SP)      ;
 2346  014370  000205                          RTS     R5              ;RETURN
```



```
 2347                                   ;BINARY TO DECIMAL ASCII CONVERT ROUTINE,
 2348                                   ;CONVERTS ARGUMENT ON STACK TO DECIMAL ASCII, STORES IT
 2349                                   ;STARTING AT ADDRESS GIVEN AFTER THE CALL,
 2350                                   ;REMOVES ARGUMENT FROM STACK BEFORE RETURNING
 2351
 2352  014372  004467  000342          BDCNV:  JSR     R4,SAV04        ;SAVE REGS 0-4,
 2353  014376  016601  000014                  MOV     12.(SP),R1      ;GET BIN VALUE,
 2354  014402  012700  026513                  MOV     #DECVAL,R0      ;GET ADDR OF DECVAL STRING,
 2355  014406  012702  014472                  MOV     #TENPWR,R2      ;ADDR OF TENPWR TO R2,
 2356  014412  012703  000005                  MOV     #5,R3           ;SET UP TO DO 5 CONVERSIONS,
 2357  014416  005004                  1$:     CLR     R4              ;CLEAR RESULT,
 2358  014420  161201                  2$:     SUB     (R2),R1         ;SUBTRACT TEN POWER,
 2359  014422  103402                          BCS     3$              ;BR IF UNSUCCESSFUL,
 2360  014424  005204                          INC     R4              ;ADD 1 TO RESULT,
 2361  014426  000774                          BR      2$              ;DO IT AGAIN,
 2362  014430  062201                  3$:     ADD     (R2)+,R1                ;RESTORE SUBTRACTED VALUE,
 2363  014432  062704  000060                  ADD     #60,R4          ;MAKE IT ASCII,
 2364  014436  110420                          MOVB    R4,(R0)+                ;SAVE IT,
 2365  014440  005303                          DEC     R3              ;DONE 5?
 2366  014442  001365                          BNE     1$              ;BR IF NOT,
 2367  014444  012501                          MOV     (R5)+,R1                ;GET FINAL STORE ADDR,
 2368  014446  012702  000005                  MOV     #5,R2           ;GET DIGIT COUNT DESIRED,
 2369  014452  060201                          ADD     R2,R1           ;COMPUTE ADDR OF 1ST DIGIT,
 2370  014454  114041                  4$:     MOVB    -(R0),-(R1)     ;TRANSFER CHAR,
 2371  014456  005302                          DEC     R2              ;DONE?
 2372  014460  001375                          BNE     4$              ;BR IF NOT,
 2373  014462  004767  000264                  JSR     PC,RST04        ;RESTORE REGS 0-4,
 2374  014466  012616                          MOV     (SP)+,(SP)
 2375  014470  000205                          RTS     R5              ;EXIT,
 2376  014472  023420  001750  000144  TENPWR: 10000.,1000.,100.,10.,1
 2377  014500  000012  000001
```

```
                                        ;RETURN TO KERNEL MODE
                                        ;
014504  042766  170000  000002 RTTK,:  BIC     #170000,2(SP)
014512  000002                         RTI
                                        ;
                                        ;
                                        ; ROUTINE TO CALC NEXT RANDOM NUMBER, USES A 29 BIT SHIFT REGISTER
                                        ; ALGORITHM, EXCLUSIVELY OR-ING THE 28TH AND 1ST(LOW ORDER) BITS
                                        ; INTO THE 29TH BIT AS IT SHIFTS RIGHT, SHIFTS 16 TIMES EACH TIME
                                        ; ITS CALLED.  MODULES DO A RAND CALL THEN JUST REFERENCE RANNUM
                                        ; WHICH IS DEFINED AS GLOBAL IN DDXCOM
                                        ;
014514  004467  000220 RAND.:  JSR     R4,SAV04        ;SAVE THE REGISTERS
014520  016700  164456         MOV     RANWRK,R0       ;GET UPPER BITS
014524  016701  164450         MOV     RANNUM,R1       ;GET LOWER ORDER 16 BITS
014530  012702  000020         MOV     #16.,R2         ;SHIFT 16 TIMES
014534  006000 RLOOP:  ROR     R0              ;SHIFT R0 INTO CARRY
014536  006001         ROR     R1              ;SHIFT CARRY INTO R1, R1 INTO CARRY
014540  103410         BCS     1$              ;FIRST HALF OF EX-OR PROCESS
                                        ;
014542  042700  010000         BIC     #10000,R0       ;ASSUME OLD BIT 28 WAS ALSO CLEAR
014546  032700  002000         BIT     #2000,R0        ;CHECK THAT ASSUMPTION
014552  001412         BEQ     2$              ;BR IF RIGHT, ELSE
014554  052700  010000         BIS     #10000,R0       ;PUT BIT IN
014560  000407         BR      2$              ;GO SHIFT AGAIN
                                        ;
014562  052700  010000 1$:    BIS     #10000,R0       ;ASSUME OLD BIT 28 WASN'T ALSO SET
014566  032700  002000         BIT     #2000,R0        ;CHECK THAT ASSUMPTION
014572  001402         BEQ     2$              ;BR IF RIGHT
014574  042700  010000         BIC     #10000,R0       ;ELSE 2 1'S = A ZERO BIT
                                        ;
014600  005302 2$:    DEC     R2              ;DONE 16 TIMES?
014602  001354         BNE     RLOOP           ;BR IF NO
014604  010067  164372         MOV     R0,RANWRK       ;SAVE CURRENT VALUES
014610  010167  164364         MOV     R1,RANNUM
014614  004767  000132         JSR     PC,RST04        ;RESTORE REG'S
014620  000002                 RTI                     ;RETURN FROM TRAP CALL
```



```
                                        ;SUBROUTINE TO GET PAGE ADDRESS FIELD VALUE (TOP 12 BITS OF ADDRESS)
                                        ;CALL VIA:
                                        ;       JSR R5,TOP12S
                                        ;       ADDRESS OF TABLE CONTAINING LOWER 16 BITS, TOP 2 BITS (SHIFTED),
                                        ;       AND SLOT FOR RETURNED VALUE OF ADDRESS
                                        ;
014622  010046 TOP12S: MOV     R0,-(SP)        ;SAVE THE OLD VALUE OF R0
014624  012500         MOV     (R5)+,R0        ;GET THE ADDRESS OF THE TABLE-PUT IT IN R0
014626  012046         MOV     (R0)+,-(SP)     ;FETCH THE ADDRESS TO BE ACTED ON TO THE STACK
014630  006216         ASR     (SP)            ;ELIMINATE THE SIX LOW ORDER BITS, THIS ONE DROPS BIT0
014632  006216         ASR     (SP)            ;THIS ROTATION DROPS BIT1
014634  006216         ASR     (SP)            ;THIS ROTATION DROPS BIT2
014636  006216         ASR     (SP)            ;THIS ROTATION DROPS BIT3
014640  006216         ASR     (SP)            ;THIS ROTATION DROPS BIT 4
014642  006216         ASR     (SP)            ;AND THIS ROTATION DROPS BIT5
014644  042716  176000         BIC     #176000,(SP)    ;ISOLATE THE 12 HIGH ORDER BITS IN THE LOWER
014650  011660  000002         MOV     (SP),2(R0)      ;SAVE THE 12 HIGH ORDER BITS IN THE TABLE'S 3RD ENTRY
014654  012016         MOV     (R0)+,(SP)      ;FETCH THE EXTENDED ADDRESS BITS FROM THE 2ND ENTRY
014656  000316         SWAB    (SP)            ;MOVE THE BITS TO BIT POSITIONS 6+7,FIRST, TO 8+9,
014660  006216         ASR     (SP)            ;THEN MOVE THEM TO 7 + 8
014662  006216         ASR     (SP)            ;FINALLY ROTATE THEM TO 6 + 7
014664  052610         BIS     (SP)+,(R0)      ;ADD THE EXTENDED ADDRESS BITS TO THE 3RD ENTRY
014666  012600         MOV     (SP)+,R0        ;RESTORE THE OLD VALUE OF R0
014670  000205         RTS     R5              ;RETURN TO THE CALLING PROCEDURE
```

```
                                        ;ROUTINE TO GET THE 18 BIT PA FROM THE 12 BIT PAF
                                        ;CALL VIA
                                        ;       JSR     R5,TO18S
                                        ;               ADDRESS OF 12 BITS
                                        ;               DEST ADDRESS OF LOW 16 BITS
                                        ;               DEST ADDRESS OF SHIFTED EA BITS

014672  010046                  TO18S:  MOV     R0,-(SP)
014674  010146                          MOV     R1,-(SP)
014676  005001                          CLR     R1
014700  013546                          MOV     @(R5)+,-(SP)
014702  012700  000006                  MOV     #6,R0
014706  006316                  1$:     ASL     (SP)
014710  006101                          ROL     R1
014712  005300                          DEC     R0
014714  001374                          BNE     1$
014716  006301                          ASL     R1
014720  006301                          ASL     R1
014722  006301                          ASL     R1
014724  006301                          ASL     R1
014726  012635                          MOV     (SP)+,@(R5)+
014730  010135                          MOV     R1,@(R5)+
014732  012601                          MOV     (SP)+,R1
014734  012600                          MOV     (SP)+,R0
014736  000205                          RTS     R5
```



```
                                        ;SAVE REGS 0-4 SUBROUTINE, (CALL VIA JSR P4,SAV04)

014740  010346                  SAV04:  MOV     R3,-(SP)        ;SAVE R3
014742  010246                          MOV     R2,-(SP)        ;SAVE R2
014744  010146                          MOV     R1,-(SP)        ;SAVE R1
014746  010046                          MOV     R0,-(SP)        ;SAVE R0
014750  010407                          MOV     R4,PC           ;R4 IS ALREADY SAVED

                                        ;RESTORE REGS 0-4 SUBROUTINE, (CALL VIA JSR PC,RST04)

014752  012604                  RST04:  MOV     (SP)+,R4        ;RETURN ADDRESS
014754  012600                          MOV     (SP)+,R0        ;RESTORE R0
014756  012601                          MOV     (SP)+,R1        ;R1
014760  012602                          MOV     (SP)+,R2        ;R2
014762  012603                          MOV     (SP)+,R3
014764  000204                          RTS     R4              ;RESTORE R4 AND RETURN
```

```
                                        ;CLEAR VARIABLES AND QUEUES, AND LOAD TRAP CATCHER STARTING AT 70

2483  014766  012700  001006            CLRQUS: MOV     #IOQUE,R0       ;CLEAR VARIABLES.
2484  014772  012701  000154                    MOV     #TKS-IOQUE,R1
2485  014776  105020                    1$:     CLRB    (R0)+
2486  015000  005301                            DEC     R1
2487  015002  001375                            BNE     1$
2488  015004  012700  027252                    MOV     #IOQ,R0         ;INITIALIZE QUEUE POINTERS
2489  015010  010067  164060                    MOV     R0,IOQ1
2490  015014  010067  164056                    MOV     R0,IOQ2
2491  015020  012767  030072  164052            MOV     #TYPEQ,TYPQ1
2492  015026  012767  030072  164046            MOV     #TYPEQ,TYPQ2
2493  015034  012701  001750                    MOV     #IOQL+TYPQL,R1  ;SET UP TO CLEAR QUEUES.
2494  015040  105020                    2$:     CLRB    (R0)+
2495  015042  005301                            DEC     R1              ;DONE?
2496  015044  001375                            BNE     2$              ;BR IF NOT.
2497  015046  000207                            RTS     PC
```



```
                                        ;LOAD TRAPCATCHER IN VECTOR AREA

2500  015050  013746  000200            CLRVEC: MOV     @#200,-(SP)
2501  015054  013746  000202                    MOV     @#202,-(SP)
2502  015060  012700  000072                    MOV     #72,R0          ;FILL VECTOR AREA WITH .+2
2503  015064  012701  000070                    MOV     #70,R1          ;AND HALT.
2504  015070  010021                    1$:     MOV     R0,(R1)+
2505  015072  005021                            CLR     (R1)+
2506  015074  010100                            MOV     R1,R0
2507  015076  005720                            TST     (R0)+
2508  015100  020027  001002                    CMP     R0,#1002        ;FILLED UP?
2509  015104  001371                            BNE     1$              ;BR IF NOT.
2510  015106  012767  015672  163000            MOV     #PARERR,PARVCT  ;RESTORE PARITY VECTOR
2511  015114  012767  000340  162774            MOV     #PRTY7,PARVCT+2 ;AND THE PRIORTY
2512  015122  012637  000202                    MOV     (SP)+,@#202
2513  015126  012637  000200                    MOV     (SP)+,@#200     ;PRESERVE   LP VECTOR
2514  015132  000207                            RTS     PC              ;YES. EXIT.

                                        ;TRAP INTERPRETER ROUTINE.

2518  015134  010046                    TRPINT: MOV     R0,-(SP)                ;PUSH R0.
2519  015136  016600  000002                    MOV     2(SP),R0                ;GET TRAP PC.
2520  015142  014000                            MOV     -(R0),R0                ;GET TRAP CALL.
2521  015144  006300                            ASL     R0              ;MULTIPLY BY 2.
2522  015146  062700  004176                    ADD     #TRPTAB-TRP2,R0 ;FORM TABLE ADDRESS
2523  015152  020027  015246                    CMP     R0,#TRPLIM      ;WITHIN LIMITS?
2524  015156  103002                            BHIS    1$              ;BR IF NOT.
2525  015160  011000                            MOV     @R0,R0          ;GET ROUTINE ADDRESS
2526  015162  000200                            RTS     R0              ;GO TO ROUTINE. RESTORE R0.
2527  015164  104406  023040            1$:     MSG$,INVTRP     ;ERROR INVALID TRAP CALL. CRASH SYSTEM.
2528  015170  104406  023746                    MSG$,HLT        ;'HALT'
2529  015174  000000                            HALT
2530          011000                            TRP2=11000
2531          000000                            TRAPX=0
2532  015176                            TRPTAB:
2533  015176                                    TRPDEF  EXIT$,EXIT.
2534  015176  005736                            EXIT.                   ;POINTER FOR TRAP CALL EXIT$
2535  015200                                    TRPDEF  QUE$,QUE.
2536  015200  006376                            QUE.                    ;POINTER FOR TRAP CALL QUE$
2537  015202                                    TRPDEF  ENDPS$,TYPQ.
2538  015202  006232                            TYPQ.                   ;POINTER FOR TRAP CALL ENDPS$
2539  015204                                    TRPDEF  END$,TYPQ.
2540  015204  006232                            TYPQ.                   ;POINTER FOR TRAP CALL END$
2541  015206                                    TRPDEF  ERROR$,TYPQ2.
2542  015206  006156                            TYPQ2.                  ;POINTER FOR TRAP CALL ERROR$
2543  015210                                    TRPDEF  DATER$,TYPQ2.
2544  015210  006156                            TYPQ2.                  ;POINTER FOR TRAP CALL DATER$
2545  015212                                    TRPDEF  MSG$,TYPQ1.
2546  015212  006244                            TYPQ1.                  ;POINTER FOR TRAP CALL MSG$
2547  015214                                    TRPDEF  BREAK$,LDBRK.
2548  015214  006366                            LDBRK.                  ;POINTER FOR TRAP CALL BREAK$
2549  015216                                    TRPDEF  ERRN$,TYPQ2.
2550  015216  006156                            TYPQ2.                  ;POINTER FOR TRAP CALL ERRN$
2551  015220                                    TRPDEF  MSGN$,TYPQ.
2552  015220  006232                            TYPQ.                   ;POINTER FOR TRAP CALL MSGN$
2553  015222                                    TRPDEF  GWBUF$,GWBUF.
```

```
2554  015222  021150                          GWBUF.                  ;POINTER FOR TRAP CALL GWBUFS
2555  015224                                  TRPDEF  GETPAS,GETPA.
2556  015224  021272                          GETPA.                  ;POINTER FOR TRAP CALL GETPAS
2557  015226                                  TRPDEF  CDATAS,CDATA.
2558  015226  007636                          CDATA.                  ;POINTER FOR TRAP CALL CDATAS
2559  015230                                  TRPDEF  MAP22S,MAP22.
2560  015230  021426                          MAP22.                  ;POINTER FOR TRAP CALL MAP22S
2561  015232                                  TRPDEF  MSGSS,TYPQ.
2562  015232  006232                          TYPQ.                   ;POINTER FOR TRAP CALL MSGSS
2563  015234                                  TRPDEF  DATCKS,DATCK.
2564  015234  007626                          DATCK.                  ;POINTER FOR TRAP CALL DATCKS
2565  015236                                  TRPDEF  RANDS,RAND.
2566  015236  014514                          RAND.                   ;POINTER FOR TRAP CALL RANDS
2567  015240                                  TRPDEF  DERRXS,TYPQ3.
2568  015240  006056                          TYPQ3.                  ;POINTER FOR TRAP CALL DERRXS
2569  015242                                  TRPDEF  MSGDS,TYPQ4.
2570  015242  006010                          TYPQ4.                  ;POINTER FOR TRAP CALL MSGDS
2571  015244                                  TRPDEF  RTTKS,RTTK.
2572  015244  014504                          RTTK.                   ;POINTER FOR TRAP CALL RTTKS
2573  015246                          TRPLIM:
```



```
2574                                    ;BUS ERROR AND RESERVED INSTRUCTION TRAP ROUTINES
2575                                    ;CHECK MODE BITS OF PSW ON STACK (MAYBE USER IF KT11 IN USE)
2576                                    ;ORDER OF ARGUMENTS: VIRTUAL PC ON STACK, PSW ON STACK, STACK POINTER
2577                                    ;AFTER TRAP, VIRTUAL ADDRESS TRAPPED TO
2578
2579  015246  012746  000004          BUSERR: MOV     #4,-(SP)                ;INDICATE BUS ERROR TRAP.
2580  015252  000413                          BR      RESIA
2581  015254  042737  000001  177572  KTERR:  BIC     #1,@#SSR0       ;TURN OFF KT11
2582  015262  104406  024050                  MSG$,KTFATL             ;'KT11 ABORT - FATAL'
2583  015266  104406  023746                  MSG$,HLT                ;'HALT'
2584  015272  000000                          HALT
2585  015274  000776                          BR      .-2             ;KT11 ABORT- FATAL ERROR
2586                                                                  ;EXAMINE KT11 SSR0 TO DETERMINE TYPE OF ABORT,
2587                                                                  ;MODE, AND PAGE REFERENCED
2588                                                                  ;EXAMINE KT11 SSR2 TO DETERMINE VIRTUAL ADDRESS
2589                                                                  ;OF INSTRUCTION WHICH ABORTED
2590  015276  012746  000010          RESINT: MOV     #10,-(SP)       ;INDICATE RESERVED INSTRUCTION TRAP.
2591  015302  000240                  RESIA:  NOP
2592  015304  010605                          MOV     SP,R5           ;SAVE THE STACK POINTER.
2593  015306  012706  027252                  MOV     #SPSPEC,SP      ;PUT INFORMATION INTO SPECIAL MONITOR STACK
2594  015312  012546                          MOV     (R5)+,-(SP)     ;PUSH TRAP VECTOR ON STACK
2595  015314  010546                          MOV     R5,-(SP)        ;PUSH STACK POINTER ON STACK
2596  015316  016546  000002                  MOV     2(R5),-(SP)     ;PUSH THE PSW ON STACK
2597  015322  011546                          MOV     (R5),-(SP)      ;PUSH THE PC ON STACK
2598  015324  012706  027212                  MOV     #SPBOT,SP       ;RESTORE MONITOR STACK POINTER
2599  015330  004767  002152                  JSR     PC,SAVLOG       ;GO SAVE 11/60 LOG  *************************
2600  015334  004767  002774                  JSR     PC,CLKOFF       ;TURN OFF CLOCK AND SAVE STATUS
2601  015340  004767  171640                  JSR     PC,CTRLX        ;CLEAR QUEUES, CLEAN UP, MAP TO BANKS 0+1
2602  015344  105267  163714                  INCB    SYSCNT          ;COUNT A SYSTEM ERROR
2603  015350  105767  163751                  TSTB    MOVING
2604  015354  001005                          BNE     1$
2605  015356  105767  163722                  TSTB    KTPRES
2606  015362  001402                          BEQ     1$
2607  015364  004767  004104                  JSR     PC,MVCODE       ;MAP BACK TO CURRENT COPY
2608  015370  012706  027242          1$:     MOV     #SPSPEC-10,SP   ;SET UP THE SPECIAL STACK POINTER
2609  015374  005067  163460                  CLR     IOBKID          ;ALLOW TYPING
2610  015400  112767  000001  163405          MOVB    #1,BRAKE        ;PREVENT MODULE STARTUP DURING MESSAGE
2611  015406  104406  023601                  MSG$,SYSERR             ;TYPE SYS ERROR FAILURE.
2612  015412  012705  013426                  MOV     #ITOA,R5
2613  015416  004715                          JSR     PC,(R5)         ;TYPE ERR PC.
2614  015420  004715                          JSR     PC,(R5)         ;TYPE ERR PSW.
2615  015422  004715                          JSR     PC,(R5)         ;TYPE SP AFTER ERROR
2616  015424  004715                          JSR     PC,(R5)         ;TYPE TRAP VECTOR ADDRESS
2617
2618  015426  012706  027212                  MOV     #SPBOT,SP       ;RESTORE MONITOR STACK POINTER
2619  015432  132767  000002  163700          BITB    #BIT1,PPRES     ;IS IT AN 11/60 ?     ******************************
2620  015440  001404                          BEQ     2$              ;NO, CONTINUE
2621  015442  104406  023420                  MSG$,SYSER1             ;YES, TELL WHAT LOG REGS, ARE PRINTED
2622  015446  004767  000360                  JSR     PC,PAMSG1       ;GO FINISH UP MESSAGE
2623  015452  104406  022737          2$:     MSG$,DOT                ;TYPE CRLF AND DOT
2624  015456  000435                          BR      PWRUA           ;ATTEMPT TO CONTINUE
```

```
                                        ;POWER DOWN AND POWER UP ROUTINES.
                                        ;
015460  004767  002650          PWRDN:  JSR     PC,CLKOFF       ;TURN OFF CLOCK AND SAVE STATUS
015464  105767  163614                  TSTB    KTPRES          ;KT11 IN USE ?
015470  001402                          BEQ     1$              ;NO, CONTINUE
015472  005037  177572                  CLR     @#SSR0          ;YES, TURN IT OFF
015476  012767  015506  162320  1$:     MOV     #PWRUP,PWRFV    ;SET UP POWER UP VECTOR,
015504  000000                          HALT                    ;POWER DOWN HALT
015506  012767  015460  162310  PWRUP:  MOV     #PWRDN,PWRFV    ;SET UP POWER DOWN VECTOR,
015514  012706  027212                  MOV     #SPBOT,R6       ;RESET STACK,
015520  005267  163542                  INC     PWRCNT          ;COUNT A POWER FAIL
015524  105067  163576                  CLRB    RUNRFL          ;CLEARRESUME RUN FLAG
015530  004767  171450                  JSR     PC,CTRLX        ;CLEAR QUEUES AND INITIALIZE KT11 IF IN USE,
015534  005067  163320                  CLR     IOBKID          ;ALLOW TYPING
015540  112767  000001  163245          MOVB    #1,BRAKE        ;PREVENT MODULE STARTUP
015546  104406  023702                  MSG$,PWRFAI             ;TYPE POWER FAILURE MESSAGE,
015552                          PWRUA:
015552  105767  163526                  TSTB    KTPRES          ;KT11 IN USE ?
015556  001412                          BEQ     1$              ;NO, CONTINUE
015560  042737  000001  177572          BIC     #1,@#SSR0       ;YES, BACK TO LOW BANK
015566  116701  163472                  MOVB    SYSCNT,R1       ;SAVE THE SYSTEM ERROR COUNT
015572  052737  000001  177572          BIS     #1,@#SSR0       ;BACK TO HIGHER BANK
015600  110167  163460                  MOVB    R1,SYSCNT       ;PUT SYS ERR COUNT IN THIS MEMORY BANK
015604  126767  163540  163452  1$:     CMPB    SYSLIM,SYSCNT   ;REACHED THE SYS, ERR, LIMIT ?
015612  101004                          BHI     2$              ;NO, CONTINUE
015614  104406  024145                  MSG$,SYSBAD             ;"EXERCISE ABORTED"
015620  000167  171220                  JMP     CTRLCA          ;CTRL C AND KILL THE EXERCISE
015624  132767  000002  163506  2$:     BITB    #BIT1,PPRES     ;IS THIS AN 11/60 ?   ****************************
015632  001413                          BEQ     3$              ;NO, CONTINUE
015634  012767  100001  163632          MOV     #100001,LGHOLD  ;YES, LOAD ERROR LOG ENABLED BIT
015642                                  WRITE   RDWHMI,WTWHMI   ;WPITE WHAMI REGISTER
015642  010046                          MOV     R0,-(SP)        ;SAVE REG, 0
015644  076600  000022                  MED,RDWHMI              ;SAVE
015650  056700  163620                  BIS     LGHOLD,R0
015654  076600  000222                  MED,WTWHMI
015660  012600                          MOV     (SP)+,R0        ;RESTORE REG, 0
015662  105767  163423          3$:     TSTB    RMODE           ;WERE WE IN RUN MODE?
015666  001126                          BNE     LOGICA          ;BR IF YES TO RESTART,
015670  000505                          BR      CHNOUT          ;NO, CONTINUE
```



```
                                        ;PARITY ERROR TRAP HANDLER ROUTINE
                                        ;
015672  022626          PARERR: CMP     (SP)+,(SP)+             ;POP THE STACK POINTER TWICE
015674  004767  002434                  JSR     PC,CLKOFF       ;MAKE SURE SYSTEM CLOCK IS OFF
015700  132767  000001  163432          BITB    #BIT0,PPRES             ;IS PARITY PRESENT/ENABLED ?
015706  001002                          BNE     2$              ;YES, CONTINUE
015710  000000                  1$:     HALT                    ;NO PARITY BUT TRAPPED ANYWAY
                                        ;RUN PARITY DIAGNOSTICS
015712  000776                          BR      1$              ;KEEP HALTING IF HE HITS CONTINUE
                                        ;
015714  132767  000002  163416  2$:     BITB    #BIT1,PPRES     ;IS IT AN 11/60 ?   ****************************
015722  001405                          BEQ     3$              ;NO, CONTINUE
015724  104406  025427                  MSG$,PERR2              ; "PARITY ERROR"
015730  004767  000076                  JSR     PC,PAMSG1       ;GO FINISH PARITY ERROR MESSAGE
015734  000706                          BR      PWRUA           ;ATTEMPT TO KEEP GOING
                                        ;
015736  012700  001430          3$:     MOV     #PARTAB,R0      ;GET ADDRESS OF PARITY ADDRESSES
015742  005001                          CLR     R1              ;ZERO THE REGISTER COUNTER
015744  005770  000000          4$:     TST     @(R0)           ;ERROR BIT SET ?
015750  100407                          BMI     6$              ;YES, GO REPORT IT
015752  005720                          TST     (R0)+           ;STEP TO THE NEXT REGISTER ADDRESS
015754  005201                          INC     R1              ;STEP THE REGISTER COUNTER
015756  126701  163357                  CMPB    CNT1,R1         ;ALL REGISTERS CHECKED ?
015762  003365                          BGT     3$              ;NO, CONTINUE
015764  000000                  5$:     HALT                    ;YES, TRAPPED BUT NO ERRORS SET
                                        ;RUN PARITY DIAGNOSTICS
015766  000776                          BR      5$              ;KEEP HALTING IF HE HITS CONTINUE
                                        ;
015770  004767  000002          6$:     JSR     PC,PAMSG        ;GO PRINT PARITY ERROR MESSAGE
015774  000666                          BR      PWRUA           ;ATTEMPT TO KEEP GOING
```

```
                                        ;PARITY ERROR MESSAGE ROUTINE, ASSUMES THAT THE ADDRESS OF THE FAILING
                                        ;    PARITY REGISTER IS IN R0.
                                        
016776  011046                  PAMSG:  MOV     (R0),-(SP)      ;GET ADDRESS OF FAILING PARITY REG.
016000  012746  000060                  MOV     #60,-(SP)       ;LOAD THE EA BITS
016004  004567  176142                  JSR     R5,OACNVX       ;CONVERT IT TO OCTAL ASCII
016010  025350                          PREG                    ;PUT IT INTO LOC. PREG
016012  017046  000000                  MOV     @(R0),-(SP)     ;GET CONTENTS OF FAILING PARITY REG.
016016  004567  176116                  JSR     R5,OACNV        ;CONVERT IT TO OCTAL ASCII
016022  025357                          PCONT                   ;PUT IT INTO LOC. PCONT
016024  104406  025330                  MSG$,PERR1              ;"PARITY ERROR -- RUN PARITY DIAG."
016030  000207                          RTS     PC              ;RETURN
                                        
                                        
016032  004767  001450          PAMSG1: JSR     PC,SAVLOG       ;SAVE ERROR LOG IN BUFFER   ************************
016036  012702  031622                  MOV     #LOGBUF+200,R2  ;GET ADR. OF SERVICE REGISTER (100*2)
016042  012701  000010                  MOV     #10,R1          ;SET THE COUNTER
016046  012246                  1$:     MOV     (R2)+,-(SP)     ;PUT CONTENTS OF LOG REG. ON STACK
016050  004567  176064                  JSR     R5,OACNV        ;CONVERT IT TO OCTAL
016054  025675                          HOLD6                   ;AND PUT IT HERE
016056  012746  025675                  MOV     #HOLD6,-(SP)    ;PUT IT ON THE STACK
016062  004767  171300                  JSR     PC,TYPE         ;AND TYPE IT
016066  005301                          DEC     R1              ;ALL DONE ?
016070  001366                          BNE     1$              ;NO, CONTINUE
016072  012746  022772                  MOV     #ACRLF,-(SP)    ;PUT A CR AND LF ON THE STACK
016076  004767  171264                  JSR     PC,TYPE         ;AND TYPE IT
016102  000207                          RTS     PC              ;RETURN
```



```
                                        ;ROUTINE TO EXIT TO CHAIN MONITOR, OR RETURN TO KYBD RTN.
                                        ;PROGRAM SHOULD ALWAYS BE IN BANK 0 BY HERE
                                        
016104  105767  163202          CHNOUT: TSTB    CHN             ;IN CHAIN MODE?
016110  100402                          BMI     1$              ;BR IF YES.
016112  000167  172530                  JMP     COMCON          ;BACK TO KYBD SERVICE.
016116  105767  163162          1$:     TSTB    KTPRES
016122  001402                          BEQ     2$              ;BRANCH IF NO KT11
016124  005037  177572                  CLR     @#SSR0          ;TURN OFF KT11
016130  016700  161706          2$:     MOV     42,R0
016134  004710                  LOGIC:  JSR     PC,(R0)         ;RETURN TO MONITOR.
016136  000240  000240  000240          .WORD   NOP,NOP,NOP
016144  105767  163142          LOGICA: TSTB    CHN             ;IN CHAIN MODE?
016150  100023                          BPL     3$              ;BR IF NO
016152  105767  163114                  TSTB    KTSAV           ;YES, KT REALLY PRESENT?
016156  001412                          BEQ     4$              ;BR IF NOT
016160  016767  163110  163040          MOV     HCR12A,HCOR12   ;SET REAL MEMORY SIZE
016166  016767  163102  163044          MOV     HCR12A,OFFMX    ;TOP ALLOWABLE PROGRAM OFFSET
016174  105267  163113                  INCB    ROTI            ;ALLOW WBUFF ROTATION
016200  105267  163111                  INCB    RELOC           ;ALLOW CODE RELOCATION
016204  012767  000001  163100  4$:     MOV     #1,CHN          ;ONLY RETIRN TO CHAIN MONITOR ONCE
016212  016767  163054  163064          MOV     KTSAV,KTPRES    ;TELL THE TRUTH
016220  105767  163060          3$:     TSTB    KTPRES
016224  001420                          BEQ     5$
016226  032737  000001  177572  1$:     BIT     #1,@#SSR0       ;KT11 ALREADY ON?
016234  001005                          BNE     2$              ;YES-SKIP REINITIALLIZING
016236  004767  173552                  JSR     PC,KTINIT
016242  012737  000001  177572          MOV     #1,@#SSR0
                                        
                                        
016250  016767  162766  162766  2$:     MOV     OFFSET,OFFSTD
016256  004767  003212                  JSR     PC,MVCODE
016262  004767  002130                  JSR     PC,CLKON        ;TURN CLOCK ON
016266  105767  163020          5$:     TSTB    CHN
016272  001402                          BEQ     6$
016274  000167  163234                  JMP     START
016300  000167  172630          6$:     JMP     RUNRES
```

```
                                        ;SUBROUTINE TO ROTATE WRITE BUFFERS IF ROTI INDICATOR =1.
                                        ;
016304 105767 163003          ROTBUF: TSTB    ROTI            ;ROTATION ALLOWED?
016310 001557                         BEQ     ROTXT           ;BR IF NOT.
016312 105767 162776                  TSTB    ROTL            ;IS ROTATION LOCKED ?
016316 100016                         BPL     1$              ;NO, CONTINUE
016320 105767 163003                  TSTB    LOCK            ;YES, IS THE PROGRAM LOCKED ?
016324 001013                         BNE     1$              ;YES, DON'T SET STOP
016326 005367 162764                  DEC     ROTCNT          ;NO, TIME TO RELOCATE ?
016332 003146                         BGT     ROTXT           ;NO, CONTINUE
016334 105767 163012                  TSTB    NSTOP           ;ALLOW STOP TO SET ?
016340 001002                         BNE     10$             ;NO, CONTINUE
016342 105267 162456                  INCB    STOP            ;YES, SET THE STOP FLAG
016346 016767 162746 162742   10$:    MOV     ROTNUM,ROTCNT   ;RESTORE THE COUNTER
016354 105767 162444          1$:     TSTB    STOP            ;IF STOP IS SET, DON'T ROTATE
016360 001133                         BNE     ROTXT
016362 105767 162726                  TSTB    ROTL            ;ROTATE LOCKED?
016366 100530                         BMI     ROTXT           ;BR IF YES
016370 105767 162425                  TSTB    MDXCTR          ;IF COUNTER NOT INITIALIZED, DON'T ROTATE
016374 100525                         BMI     ROTXT
016376 016746 162630                  MOV     WBUF,-(SP)      ;CHECK CURRENT VALUE
016402 042716 160000                  BIC     #160000,(SP)
016406 022726 016000                  CMP     #16000,(SP)+
016412 003016                         BGT     3$              ;IF NOT LAST IN BANK, BRANCH
016414 105767 162401                  TSTB    MDXCTR          ;HAVE ALL MODULES USED A VALUE IN THIS BANK?
016420 001403                         BEQ     2$              ;YES- TIME TO CHANGE BANKS
016422 105367 162706                  DECB    $GWBFC          ;NO, DEC TIMEOUT COUNTER
016426 001110                         BNE     ROTXT           ;EXIT UNLESS TIMEOUT HAS OCCURRED
016430 116767 162366 162363   2$:     MOVB    MDXCNT,MDXCTR   ;RESET COUNTERS
016436 116767 162367 162670           MOVB    ROTCT,$GWBFC
016444 105167 162353                  COMB    WBFLG           ;TOGGLE INDICATOR
016450 026767 162562 162564   3$:     CMP     WBUF12,OFFSET   ;LOCORE ROTATION?
016456 101453                         BLOS    ROTLO1          ;YES- BRANCH
016460 062767 000020 162550           ADD     #20,WBUF12
016466 026767 162544 162532           CMP     WBUF12,HCOR12   ;CHECK FOR TOP OF WRITE BUFFER AREA
016474 103427                         BLO     ROTB2           ;IF NOT, BRANCH
016476 000400                         BR      4$              ;IF OVER, START AGAIN
016500 026727 162536 000400   4$:     CMP     OFFSET,#400     ;ROTATE THRU MEMORY BELOW PROGRAM IF ENOUGH ROOM
016506 103403                         BLO     ROTB1           ;NOT ENOUGH ROOM
016510 004767 000136                  JSR     PC,ROTLOW       ;SETUP TO ROTATE BELOW PROGRAM
016514 000455                         BR      ROTXT           ;RETURN
016516 105767 162573          ROTB1:  TSTB    RELOC           ;RELOCATION ALLOWED?
016522 001426                         BEQ     ROTB3           ;NO- BRANCH
016524 105767 162577                  TSTB    LOCK            ;LOCKED?
016530 001023                         BNE     ROTB3           ;YES- BRANCH
016532 105767 162614                  TSTB    NSTOP           ;ALLOW STOP TO SET ?
016536 001002                         BNE     1$              ;NO, CONTINUE
016540 105267 162260                  INCB    STOP            ;YES, SET STOP
016544 162767 000020 162464   1$:     SUB     #20,WBUF12      ;RESTORE WBUF
016552 000436                         BR      ROTXT           ;EXIT
016554 062767 002000 162450   ROTB2:  ADD     #2000,WBUF      ;BUMP ADDR- POINT 512 WORDS HIGHER
016562 103003                         BCC     1$
016564 062767 000020 162442           ADD     #20,WBFEA
016572 004767 173130          1$:     JSR     PC,SETWB1       ;CALCULATE MXWBF
016576 000424                         BR      ROTXT
016600 004767 173040          ROTB3:  JSR     PC,SETWBF
```



```
016604 000421                         BR      ROTXT           ;RETURN
                                        ;
016606 062767 000020 162422   ROTLO1: ADD     #20,WBUF12
016614 026767 162416 162420           CMP     WBUF12,OFFSET
016622 103401                         BLO     ROTB4           ;OK- USE IT
016624 000734                         BR      ROTB1           ;NO- RELOCATE OR START WBUF CYCLE OVER
016626 062767 002000 162376   ROTB4:  ADD     #2000,WBUF      ;ADVANCE THE WRITE BUFFER
016634 103003                         BCC     1$              ;IF NO CARRY DON'T ADJUST THE EA BITS
016636 062767 000020 162370           ADD     #20,WBFEA       ;CARRY SET, ADJUST THE EA BITS
016644 004767 000016                  JSR     PC,ROTLW1       ;TAKE CARE OF ROTATING BELOW THE PROGRAM
016650 000207                 ROTXT:  RTS     PC              ;RETURN
```

```
                                        ;THIS ROUTINE WILL HANDLE THE SITUATION WHEN THE WRITE BUFFER IS ROTATING
                                        ;    BENEITH THE PROGRAM
                                        
016652  005067  162354        ROTLOW: CLR     WBUF
016656  005067  162352                CLR     WBFEA
016662  005067  162350                CLR     WBUF12
016666  016767  162350  162362 ROTLW1: MOV     OFFSET,MXWBF
016674  166767  162336  162354        SUB     WBUF12,MXWBF
016702  006367  162350                ASL     MXWBF
016706  006367  162344                ASL     MXWBF
016712  006367  162340                ASL     MXWBF
016716  006367  162334                ASL     MXWBF
016722  006367  162330                ASL     MXWBF
016726  103003                        BCC     1$
016730  012767  177777  162320        MOV     #177777,MXWBF
016736  000207              1$:       RTS     PC              ;RETURN
                                        
                                        
                                        ;SUBROUTINE TO CHECK IF BUFFER ROTATION ENABLED, AND TO TYPE
                                        ;THE ROTATION RANGE.
                                        
016740  105767  162371        ROTCHK: TSTB    MEMERF          ;ANY MEMORY ERRORS?
016744  001406                        BEQ     1$              ;NO= BRANCH
016746  104406  026344                MSG$,MERR1
016752  104406  023724                MSG$,MEMCHG
016756  105067  162353                CLRB    MEMERF
016762  105767  162325        1$:     TSTB    ROTI            ;ROTATE ENABLED?
016766  001430                        BEQ     2$              ;BR IF NOT.
016770  104406  024505                MSG$,ROTENB
016774  016746  162220                MOV     LOCORE,-(SP)    ;TYPE LOWER LIMIT.
017000  004767  174422                JSR     PC,ITOA
017004  004567  175662                JSR     R5,TO18S        ;CONVERT UPPER LIMIT
017010  001226                        HCOR12
017012  001132                        SAVLO
017014  001134                        SAVHI
017016  162767  000002  162106        SUB     #2,SAVLO
017024  103003                        BCC     3$
017026  162767  000020  162100        SUB     #20,SAVHI
017034  016746  162072        3$:     MOV     SAVLO,-(SP)     ;TYPE UPPER LIMIT
017040  016746  162070                MOV     SAVHI,-(SP)
017044  004767  174266                JSR     PC,ITOAX
                                        
017050  000207              2$:       RTS     PC              ;EXIT.
```



```
                                        ;THIS ROUTINE PRINTS OUT THE SYSTEM MEMORY SIZE IN DECIMAL.
                                        
017052  016700  162150        SYSIZ:  MOV     HCOR12,R0       ;GET THE MEMORY SIZE
017056  006200                        ASR     R0              ;CONVERT FROM BYTE SIZE TO K WORD SIZE
017060  006200                        ASR     R0              ;
017062  006200                        ASR     R0              ;
017064  006200                        ASR     R0              ;
017066  006200                        ASR     R0              ;
017070  010046                        MOV     R0,-(SP)        ;PUT IT ON THE STACK
017072  004567  175274                JSR     R5,BDCNV        ;CONVERT TO DECIMAL ASCII CHARACTERS
017076  025667                        HOLD5                   ;AND PUT THEM HERE
017100  012700  025667                MOV     #HOLD5,R0       ;GET THE LOCATION OF THE ASCII NUMBER
017104  122720  000060        1$:     CMPB    #60,(R0)+       ;IS IT A LEADING 0?
017110  001004                        BNE     2$              ;NO, CONTINUE
017112  112760  000040  177777        MOVB    #40,-1(R0)      ;YES, FILL IT WITH A BLANK
017120  000771                        BR      1$              ;CHECK THE NEXT CHARACTER
017122  104406  024465        2$:     MSG$,SYSSIZ             ;"SYSTEM SIZE:"
017126  104406  025667                MSG$,HOLD5              ;TYPE THE SYSTEM SIZE
017132  104406  024503                MSG$,KAY                ;"K"
017136  000207                        RTS     PC              ;RETURN
```

```
                                ;       THIS ROUTINE WILL PRINT THE NUMBER OF SYSTEM ERRORS
                                ;FOUND IN COUNTER SYSCNT,       CALL VIA:       JSR     PC,SYSNUM
                                
017140  010046                  SYSNUM: MOV     R0,-(SP)        ;SAVE REG, 0
017142  010146                          MOV     R1,-(SP)        ;SAVE REG, 1
017144  016746  162114                  MOV     SYSCNT,-(SP)    ;PUT SYSTEM ERROR COUNT ON STACK
017150  004567  175216                  JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII CHARS,
017154  025667                          HOLD5                   ;AND PUT THEM HERE
017156  012701  000004                  MOV     #4,R1           ;LOAD THE COUNTER
```



```
017162  012700  025667                  MOV     #HOLD5,R0       ;GET ADR OF THE ASCII NUMBER

017166  122720  000060          1$:     CMPB    #60,(R0)+       ;IS IT A LEADING ZERO ?
017172  001005                          BNE     2$              ;NO, CONTINUE
017174  112760  000040  177777          MOVB    #40,-1(R0)      ;YES, REPLACE IT WITH A BLANK
017202  005301                          DEC     R1              ;ALL DONE ?

017204  001370                          BNE     1$              ;NO, CHECK MORE CHARACTERS

017206  104406  025531          2$:     MSG$,NUMSYS             ;"SYSTEM ERRORS:"
017212  104406  025667                  MSG$,HOLD5              ;TYPE THE NUMBER

017216  012601                          MOV     (SP)+,R1        ;RESTORE REG, 1
017220  012600                          MOV     (SP)+,R0        ;RESTORE REG, 0
017222  000207                          RTS     PC              ;RETURN
```

```
                                        ;       THIS ROUTINE WILL PRINT THE NUMBER OF POWER FAIL ERRORS
                                        ;FOUND IN COUNTER PWRCNT.       CALL VIA:       JSR     PC,PWRNUM
                                        PWRNUM: MOV     R0,-(SP)        ;SAVE REG. 0
                                                MOV     R1,-(SP)        ;SAVE REG. 1
017224  010046                                  MOV     PWRCNT,-(SP)    ;PUT POWER FAIL ERROR COUNT ON STACK
017226  010146
017230  016746  162032                          JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII CHARS.
017234  004567  175132                          HOLD5                   ;AND PUT THEM HERE
017240  025667                                  MOV     #4,R1           ;LOAD THE COUNTER
017242  012701  000004                          MOV     #HOLD5,R0       ;GET ADR OF THE ASCII NUMBER
017246  012700  025667
                                        1$:     CMPB    #60,(R0)+       ;IS IT A LEADING ZERO ?
017252  122720  000060                          BNE     2$              ;NO, CONTINUE
017256  001005                                  MOVB    #40,-1(R0)      ;YES, REPLACE IT WITH A BLANK
017260  112760  000040  177777                  DEC     R1              ;ALL DONE ?
017266  005301                                  BNE     1$              ;NO, CHECK MORE CHARACTERS
017270  001370
                                        2$:     MSG$,NUMPWR             ;"POWER FAILS:"
017272  104406  025551                          MSG$,HOLD5              ;TYPE THE NUMBER
017276  104406  025667
                                                MOV     (SP)+,R1        ;RESTORE REG. 1
017302  012601                                  MOV     (SP)+,R0        ;RESTORE REG. 0
017304  012600                                  RTS     PC              ;RETURN
017306  000207
```



```
                                        ;       THIS ROUTINE WILL CONVERT THE GIVEN BINARY
                                        ;NUMBER INTO HOURS, MINUTES, AND SECONDS.  LOCATIONS
                                        ;HOLDING THE FINAL NUMBERS WILL BE TAGGED AS:  HOURS,
                                        ;MINITS, AND SEKNDS. HOURS MUST BE ON AN ODD BOUNDARY.
                                        ;MINITS AND SEKNDS MUST BE ON EVEN BOUNDARIES.
                                        ;TEMPORARY LOCATION HOLD5 MUST BE ON AN ODD BOUNDARY.
                                        ;               CALL VIA:       JSR     PC,HMS
                                        ;                               ADR             ;ADDRESS OF THE TIME
                                        ;                               ADRX            ;ADDRESS OF EXTENDED TIME BITS
017310  004467  175424                  HMS:    JSR     R4,SAV04        ;SAVE REGS. 0-4
017314  016600  000012                          MOV     10.(SP),R0      ;GET ADR OF OF ADR TIME
017320  010003                                  MOV     R0,R3           ;COPY IT INTO REG. 3
017322  005723                                  TST     (R3)+           ;GET ADR OF EXTENDED TIME BITS ADR
017324  017000  000000                          MOV     @(R0),R0        ;PUT THE TIME IN REG. 0
017330  017303  000000                          MOV     @(R3),R3        ;PUT EXTENDED TIME BITS INTO REG. 3
017334  005001                                  CLR     R1              ;ZERO MINUTES COUNTER
017336  005002                                  CLR     R2              ;ZERO HOURS COUNTER
                                        1$:     CMP     #3600.,R0       ;LESS THAN 1 HOUR?
017340  022700  007020                          BHI     3$              ;YES, GO LOOK FOR MORE HOURS
017344  101013                                  SUB     #3600.,R0       ;NO, SUBTRACT 1 HOUR'S WORTH OF SECONDS
017346  162700  007020                          INC     R2              ;COUNT AN HOUR
017352  005202                                  BR      1$              ;CHECK FOR MORE HOURS
017354  000771
                                        2$:     CMP     #60.,R0         ;LESS THAN 1 MINUTE?
017356  022700  000074                          BHI     4$              ;YES, ALL DONE
017362  101014                                  SUB     #60.,R0         ;NO, SUBTRACT 1 MINUTE'S WORTH OF SECONDS
017364  162700  000074                          INC     R1              ;COUNT A MINUTE
017370  005201                                  BR      2$              ;CHECK FOR MORE MINUTES
017372  000771
                                        3$:     TST     R3              ;ANY EXTENDED TIME BITS ?
017374  005703                                  BEQ     2$              ;NO, GO LOOK FOR MINUTES
017376  001767                                  ADD     #17.,R2         ;YES, ADD IN 17. HOURS
017400  062702  000021                          ADD     #4336.,R0       ;ADD IN 1 HR 12 MIN 16 SEC
017404  062700  010360                          DEC     R3              ;GET RID OF ONE TIME EXTENDED BIT
017410  005303                                  BR      1$              ;CONTINUE
017412  000752
                                        4$:     MOV     R2,-(SP)        ;PUT THE HOURS ON THE STACK
017414  010246                                  JSR     R5,BDCNV        ;CONVERT IT TO ASCII
017416  004567  174750                          HOLD5                   ;AND PUT IT HERE
017422  025667                                  MOVB    HOLD5+2,HOURS   ;LOAD HOURS
017424  116767  006241  161753                  MOV     HOLD5+3,HOURS+1 ;LOAD HOURS
017432  016767  006234  161746                  MOV     R1,-(SP)        ;PUT THE MINUTES ON THE STACK
017440  010146                                  JSR     R5,BDCNV        ;CONVERT IT TO ASCII
017442  004567  174724                          HOLD5                   ;AND PUT IT HERE
017446  025667                                  MOV     HOLD5+3,MINITS  ;LOAD MINUTES
017450  016767  006216  161732                  MOV     R0,-(SP)        ;PUT THE SECONDS ON THE STACK
017456  010046                                  JSR     R5,BDCNV        ;CONVERT IT TO ASCII
017460  004567  174706                          HOLD5                   ;AND PUT IT HERE
017464  025667                                  MOV     HOLD5+3,SEKNDS  ;LOAD SECONDS
017466  016767  006200  161716
                                                JSR     PC,RST04        ;RESTORE REGS 0-4
017474  004767  175252                          ADD     #4,(SP)         ;SET PROPER RETURN PC
017500  062716  000004                          RTS     PC              ;AND RETURN
017504  000207
```

2993


```
2994                                    ;       THIS ROUTINE COPIES THE 11/60 ERROR LOG AND PUTS IT INTO
2995                                    ;A BUFFER IN THE MONITOR.
2996
2997  017506  132767  000002  161624  SAVLOG: BITB    #BIT1,PPRES     ;IS IT AN 11/60 ?   ******************************
2998  017514  001414                          BEQ     3$              ;NO, GET OUT
2999  017516  012701  031422                  MOV     #LOGBUF,R1      ;GET ADR. OF LOG BUFFER
3000  017522  005002                          CLR     R2              ;INITIALIZE LOG COMMAND AND ADDRESS
3001
3002  017524  010267  000002          1$:     MOV     R2,2$           ;LOAD READ COMMAND AND THE ADDRESS
3003  017530  076600                          MFD                     ;READ THE LOG REGISTER
3004  017532  000000                  2$:     .WORD                   ;READ COMMAND AND REGISTER ADDRESS
3005  017534  010021                          MOV     R0,(R1)+        ;STORE CONTENTS OF LOG IN BUFFER
3006  017536  005202                          INC     R2              ;STEP TO NEXT ADDRESS
3007  017540  022702  000177                  CMP     #177,R2         ;ALL DONE ?
3008  017544  001367                          BNE     1$              ;NO, KEEP GOING
3009
3010  017546  000207                  3$:     RTS     PC              ;YES, RETURN
3011
```

```
                                ;       THIS ROUTINE CHECKS TO SEE IF ANY CLOCK
                                ;MODULES ARE PRESENT AND SETS BIT FLAGS IN BYTE
                                ;LOCATION CLOCK
                                ;               CALL VIA:       JSR     PC,KLOCK
                                ;
017550  105067  161577          KLOCK:  CLRB    CLOCK           ;ZERO CLOCK FLAG
017554  005767  161576                  TST     CLOCKL          ;KW11-L CLOCK PRESENT?
017560  001406                          BEQ     1$              ;NO, CONTINUE
017562  112767  000001  161563          MOVB    #BIT0,CLOCK     ;SET L CLOCK PRESENT FLAG
017570  017700  161562                  MOV     @CLOCKL,R0      ;GET BEGIN ADDRESS OF CLOCK MODULE
017574  000410                          BR      2$              ;CONTINUE

017576  005767  161556          1$:     TST     CLOCKP          ;KW11-P CLOCK PRESENT?
017602  001421                          BEQ     4$              ;NO, RETURN
017604  112767  000002  161541          MOVB    #BIT1,CLOCK     ;SET P CLOCK PRESENT FLAG
017612  017700  161542                  MOV     @CLOCKP,R0      ;GET BEGIN ADDRESS OF CLOCK MODULE
017616  012737  [illegible]  000004  2$:     MOV     #3$,@#4         ;LOAD TIME-OUT VECTOR
017624  005070  000006                  CLR     @6(R0)          ;MAKE SURE SYSTEM CLOCK IS THERE
017630  000406                          BR      4$              ;IT'S THERE, RETURN OK

017632  022626                  3$:     CMP     (SP)+,(SP)+             ;POP THE STACK POINTER TWICE
017634  104406  024317                  MSG$,NOCLK              ;'NO SYSTEM CLOCK FOUND'
017640  000000                          HALT                    ;WAIT FOR OPERATOR'S DECISION
017642  105067  161505                  CLRB    CLOCK           ;INDICATE NO CLOCK AVAILABLE
017646  012737  015246  000004  4$:     MOV     #BUSERR,@#4     ;RESTORE TIME-OUT VECTOR
017654  000207                          RTS     PC              ;RETURN
```



```
                                ;       THIS ROUTINE WILL GET THE RUNTIME IN
                                ;SECONDS FROM THE CLOCK MODULE, HAVE IT CONVERTED
                                ;TO HOURS, MINUTES, AND SECONDS VIA SUBROUTINE HMS, AND
                                ;STUFFS IT INTO PROPER LOCATION FOR PRINTOUT,
                                ;               CALL VIA:       JSR     PC,RUNTYM

017656  010046                  RUNTYM: MOV     R0,-(SP)        ;SAVE REG, 0
017660  132767  000001  161465          BITB    #BIT0,CLOCK     ;USING THE KW11-L CLOCK?
017666  001005                          BNE     1$              ;YES, CONTINUE
017670  132767  000002  161455          BITB    #BIT1,CLOCK     ;USING THE KW11-P CLOCK?
017676  001004                          BNE     2$              ;YES, CONTINUE
017700  000456                          BR      5$              ;NO, CAN'T DO ANYTHING, RETURN

017702  016700  161450          1$:     MOV     CLOCKL,R0       ;GET ADR OF CLOCK INFORMATION

017706  000402                          BR      3$              ;CONTINUE
017710  016700  161444          2$:     MOV     CLOCKP,R0       ;GET ADR OF CLOCK INFORMATION

017714  016067  000002  161472  3$:     MOV     2(R0),RTIME     ;GET THE TIME FROM THE CLOCK MODULE
017722  016067  177776  161466          MOV     -2(R0),RTIMX    ;GET THE TIME EXTENDED BITS TOO
017730  105767  161355                  TSTB    RMODE           ;PROGRAM IN THE RUN MODE ?
017734  001407                          BEQ     4$              ;NO, MUST BE TIME FOR SUMMARY
017736  105767  161407                  TSTB    TFLAG           ;YES, IS IT ERROR TIME ?
017742  100404                          BMI     4$              ;YES, SETUP RUNTIME FOR OUTPUT
017744  032777  010000  161224          BIT     #BIT12,@SP      ;NO, PRINT THE EOP DATA ?

017752  001431                          BEQ     5$              ;NO, RETURN

017754  004767  177330          4$:     JSR     PC,HMS          ;CONVERT TO HOURS, MINUTES AND SECONDS
017760  001414                          RTIME                   ;FROM SECONDS
017762  001416                          RTIMX                   ;EXTENDED BITS
017764  116767  161415  006105          MOVB    HOURS,THRS      ;LOAD 100'S COLUMN OF HOURS
017772  116767  161410  006100          MOVB    HOURS+1,THRS+1  ;LOAD 10'S COLUMN OF HOURS
020000  116767  161403  006073          MOVB    HOURS+2,THRS+2  ;LOAD 1'S COLUMN OF HOURS
020006  116767  161376  006067          MOVB    MINITS,TMINS    ;LOAD 10'S COLUMN OF MINUTES
020014  116767  161371  006062          MOVB    MINITS+1,TMINS+1  ;LOAD 1'S COLUMN OF MINUTES
020022  116767  161364  006056          MOVB    SEKNDS,TSECS    ;LOAD 10'S COLUMN OF SECONDS
020030  116767  161357  006051          MOVB    SEKNDS+1,TSECS+1  ;LOAD 1'S COLUMN OF SECONDS

020036  012600                  5$:     MOV     (SP)+,R0        ;RESTORE REG, 0
020040  000207                          RTS     PC              ;RETURN
```

```
                                        ;       THIS ROUTINE WILL SETUP THE TIME INFORMATION
                                        ;FOR THE END OF PASS AND ERROR MESSAGES, IT ALSO LOADS THE START
                                        ;TIME FOR A BKMOD.  SUBROUTINES CALLED ARE RUNTYM AND HMS.
                                        ;               CALL VIA:       JSR     PC,PASTYM
                                        ;                               JSR     PC,ERRTYM
                                        ;                               JSR     PC,BKTYM

020042  112767  000001  161301  BKTYM:  MOVB    #BIT0,TFLAG     ;SET FLAG INDICATING BKMOD TIME SETUP
020050  000406                          BR      RUNPAS          ;CONTINUE
020052  112767  000200  161271  ERRTYM: MOVB    #BIT7,TFLAG     ;SET ERROR CALL FLAG
020060  000402                          BR      RUNPAS          ;CONTINUE
020062  105067  161263          PASTYM: CLRB    TFLAG           ;SET END OF PASS FLAG

020066  004467  174646          RUNPAS: JSR     R4,SAV04        ;SAVE REGS, 0-4
020072  132767  000001  161253          BITB    #BIT0,CLOCK     ;USING THE KW11-L CLOCK?
020100  001005                          BNE     1$              ;YES, CONTINUE
020102  132767  000002  161243          BITB    #BIT1,CLOCK     ;USING THE KW11-P CLOCK?
020110  001004                          BNE     2$              ;YES, CONTINUE
020112  000505                          BR      9$              ;NO, CAN'T DO ANYTHING, RETURN

020114  016700  161236          1$:     MOV     CLOCKL,R0       ;GET ADR OF CLOCK INFORMATION
020120  000402                          BR      3$              ;CONTINUE
020122  016700  161232          2$:     MOV     CLOCKP,R0       ;GET ADR OF CLOCK INFORMATION

020126  132767  000001  161215  3$:     BITB    #BIT0,TFLAG     ;SETUP FOR BKMOD ?
020134  001403                          BEQ     4$              ;NO, CONTINUE
020136  017701  161126                  MOV     @BKPTR,R1       ;YES, GET ADR OF THE BKMOD
020142  000402                          BR      5$              ;CONTINUE

020144  016701  161344          4$:     MOV     CSTART,R1       ;GET ADR OF MODULE DOING EOP
020150  012702  026522          5$:     MOV     #MODQ,R2        ;GET ADR OF MODULE ADDRESSES
020154  012703  000004                  MOV     #4,R3           ;SET POINTER TO 1ST MODULE TIME ENTRY

020160  020122                  6$:     CMP     R1,(R2)+        ;HAS THE MODULE TIME BEEN FOUND?
020162  001406                          BEQ     7$              ;YES, CONTINUE
020164  062703  000004                  ADD     #4,R3           ;ADD 4 TO THE TABLE ADDRESS
020170  022703  000360                  CMP     #240.,R3        ;AT THE END OF THE TABLE?
020174  003371                          BGT     6$              ;NO, KEEP SEARCHING
020176  000453                          BR      9$              ;SOMETHING'S WRONG--GET OUT

020200  060003                  7$:     ADD     R0,R3           ;GET ADR OF CLOCK INFO TABLE
020202  011304                          MOV     (R3),R4         ;GET TIME OF MODULE'S LAST PASS
020204  004767  177446                  JSR     PC,RUNTYM       ;GET AND SET UP RUNTIME INFORMATION
020210  016700  161200                  MOV     RTIME,R0        ;PUT THE RUNTIME IN REG, 0
020214  160400                          SUB     R4,R0           ;GET THE MODULE'S ELAPSED TIME
020216  105767  161127                  TSTB    TFLAG           ;IS IT ERROR TIME ?
020222  100415                          BMI     8$              ;YES, CONTINUE
020224  016713  161164                  MOV     RTIME,(R3)      ;NO, UPDATE THE MODULE'S PASS TIME
020230  016763  161162  000002          MOV     RTIMX,2(R3)     ;AND THE EXTENDED BITS

020236  132767  000001  161105          BITB    #BIT0,TFLAG     ;SETUP FOR BKMOD ?
020244  001030                          BNE     9$              ;YES, ALL DONE, RETURN
020246  032777  010000  160722          BIT     #BIT12,@SR      ;NO, PRINT THE EOP DATA ?
020254  001424                          BEQ     9$              ;NO, RETURN
020256  010067  161142          8$:     MOV     R0,PTIME        ;PUT THE ELAPSED TIME INTO PTIME
```



```
020262  005067  161130                  CLR     RTIMX           ;THIS WILL ALWAYS BE 0
020266  004767  177016                  JSR     PC,HMS          ;CONVERT IT TO HOURS, MINUTES, AND SECONDS
020272  001424                          PTIME                   ;GIVEN THE TIME IN SECONDS
020274  001416                          RTIMX                   ;0
020276  116767  161106  005615          MOVB    MINITS,MMINS    ;LOAD 10'S COLUMN OF MINUTES
020304  116767  161101  005610          MOVB    MINITS+1,MMINS+1  ;LOAD 1'S COLUMN OF MINUTES
020312  116767  161074  005604          MOVB    SEKNDS,MSECS    ;LOAD 10'S COLUMN OF SECONDS
020320  116767  161067  005577          MOVB    SEKNDS+1,MSECS+1 ;LOAD 1'S COLUMN OF SECONDS

020326  004767  174420          9$:     JSR     PC,RST04        ;RESTORE REGS, 0-4
020332  000207                          RTS     PC              ;RETURN
```

```
                                ;       THIS ROUTINE WILL SHUT OFF THE CLOCK (IF AVAILABLE) TO KEEP
                                ;IT FROM INTERUPTING DURING THE MVCODE ROUTINE.
                                ;                       CALL VIA:       JSR     PC,CLKOFF

020334  105767  161013        CLKOFF: TSTB    CLOCK           ;IS THERE A CLOCK ON THE SYSTEM ?
020340  001425                        BEQ     4$              ;NO, RETURN
020342  132767  000001  161003        BITB    #BIT0,CLOCK     ;YES, THE KW11-L ?
020350  001004                        BNE     1$              ;YES, CONTINUE
020352  132767  000002  160773        BITB    #BIT1,CLOCK     ;NO, HOW ABOUT THE KW11-P ?
020360  001003                        BNE     2$              ;YES, CONTINUE

020362  017700  160770          1$:   MOV     @CLOCKL,R0      ;GET THE BEGIN ADDRESS OF CLOCK MODULE
020366  000402                        BR      3$              ;CONTINUE
020370  017700  160764          2$:   MOV     @CLOCKP,R0      ;GET THE BEGIN ADDRESS OF CLOCK MODULE

020374  016067  000006  161016  3$:   MOV     6(R0),CLKADR    ;SAVE THE CLOCK ADDRESS
020402  017067  000006  161012        MOV     @6(R0),CLKHLD   ;SAVE CONTENTS OF CLOCK CSR
020410  005070  000006                CLR     @6(R0)          ;KILL THE CLOCK
020414  000207                  4$:   RTS     PC              ;RETURN


                                ;       THIS ROUTINE WILL TURN THE CLOCK BACK ON IF IT WAS TURNED
                                ;OFF BY CLKOFF.
                                ;                       CALL VIA:       JSR     PC,CLKON

020416  105767  160731        CLKON:  TSTB    CLOCK           ;IS THERE A CLOCK ON THE SYSTEM ?
020422  001403                        BEQ     1$              ;NO, RETURN
020424  016777  160772  160766        MOV     CLKHLD,@CLKADR  ;RESTORE THE CLOCK STATUS
020432  000207                  1$:   RTS     PC              ;RETURN


                                ;       THIS IS THE CALLING SEQUENCE FOR THE CKHUNG ROUTINE

020434  016706  160512        HUNGCK: MOV     SPSAV,SP        ;RESTORE MONITOR STACK POINTER
020440  005067  157332                CLR     PSW             ;PROCESSOR TO STATUS 0
020444  005067  161044                CLR     CSTART          ;INDICATE NO MODULES DOING EOP
020450  004767  000006                JSR     PC,CKHUNG       ;GO CHECK FOR HUNG MODULES
020454  000240                        NOP
020456  000167  164476                JMP     ENDSVG          ;GO CHECK FOR OTHER MODULES RUNNING
```



```
                                ;*************************************************************************
                                ;THIS ROUTINE IS CALLED FROM THE "PASEND" OR HUNGCK ROUTINE AND CHECKS
                                ;THAT ALL RUNNING MODULES HAVE COMPLETED AT LEAST ONE PASS DURING A
                                ;SPECIFIED TIME INTERVAL. IF IT DETECTS A "HUNG" MODULE, THE
                                ;MODULE IS DROPPED AND AN ERROR MESSAGE REPORTED.

020462  004467  174252        CKHUNG: JSR     R4,SAV04        ;GO SAVE R0 - R4
020466  132767  000001  160657        BITB    #BIT0,CLOCK     ;USING THE KW11-L CLOCK ?
020474  001007                        BNE     1$              ;BR IF YES
020476  132767  000002  160647        BITB    #BIT1,CLOCK     ;USING THE KW11-P CLOCK ?
020504  001513                        BEQ     7$              ;BR IF NOT - SOMETHING'S SCREWEY ??
020506  016702  160646                MOV     CLOCKP,R2       ;GET POINTER TO TIMING INFO
020512  000402                        BR      2$              ;CONTINUE ON
020514  016702  160636          1$:   MOV     CLOCKL,R2       ;GET POINTER TO TIMING INFO
020520  010203                  2$:   MOV     R2,R3           ;COPY R2
020522  062703  000004                ADD     #4,R3           ;R3 POINTS TO TABLE OF PASS TIMES
                                                              ;IN THE "TIMER" MODULE
020526  012700  026522                MOV     #MODQ,R0        ;GET POINTER TO MODULE ADDR TABLE
020532  005067  160670                CLR     MAXTIM          ;ZERO LARGEST ELAPSED TIME
020536  012001                  3$:   MOV     (R0)+,R1        ;PUT MODULE BEGIN ADDR IN R1
020540  001442                        BEQ     4$              ;BR IF DONE ALL MODULES
020542  026701  160746                CMP     CSTART,R1       ;IS IT THE MODULE DOING THE EOP ?
020546  001467                        BEQ     6$              ;YES, SKIP IT
020550  022761  144000  000020        CMP     #144000,STAT(R1) ;IS IT AN IOMODR ?
020556  001463                        BEQ     6$              ;BR IF YES
020560  032761  040000  000020        BIT     #BIT14,STAT(R1) ;IS IT SELECTED ??
020566  001457                        BEQ     6$              ;BR IF NOT
020570  032761  020000  000020        BIT     #BIT13,STAT(R1) ;WAS IT ALREADY DROPPED ??
020576  001053                        BNE     6$              ;BR IF YES
020600  005761  000020                TST     STAT(R1)        ;IS IT A BKMOD ?
020604  100050                        BPL     6$              ;YES, FORGET IT
020606  022761  142000  000020        CMP     #142000,STAT(R1) ;IS IT AN IOMODP ?
020614  001444                        BEQ     6$              ;YES, FORGET IT
020616  016204  000002                MOV     2(R2),R4        ;GET TIME FROM THE TIMER
020622  161304                        SUB     (R3),R4         ;CALCULATE TIME SINCE LAST ENDPAS
020624  062703  000004                ADD     #4,R3           ;POINT TO THE NEXT TIME ENTRY
020630  020467  160572                CMP     R4,MAXTIM       ;IS THIS ELAPSED TIME BIGGER ?
020634  101740                        BLOS    3$              ;NO, CONTINUE
020636  010105                        MOV     R1,R5           ;YES, SAVE THE MODULE POINTER
020640  010467  160562                MOV     R4,MAXTIM       ;UPDATE THE MAXTIM
020644  000734                        BR      3$              ;CONTINUE
020646  026767  160510  160552  4$:   CMP     HNGTIM,MAXTIM   ;IS THE MODULE HUNG ??
020654  100027                        BPL     7$              ;IF NOT, RETURN
020656  010501                        MOV     R5,R1           ;YES, INDICATE MODULE TO BE DROPPED
020660  004767  000070                JSR     PC,DRPHNG       ;GO DROP IT AND REPORT ERROR
020664  105767  160134                TSTB    STOP            ;WAITING TO RELOCATE ?
020670  001421                        BEQ     7$              ;NO, RETURN
020672  012700  026522                MOV     #MODQ,R0        ;SET POINTER TO MODULE ADDR TABLE
020676  016204  000002                MOV     2(R2),R4        ;YES, GET THE PRESENT TIME
020702  016201  177776                MOV     -2(R2),R1       ;AND TIME EXTENDED BITS
020706  010203                        MOV     R2,R3           ;GET POINTER TO TIME INFO
020710  062703  000004                ADD     #4,R3           ;POINT TO PASTIME INFO

020714  005720                  5$:   TST     (R0)+           ;ALL DONE ?
020716  001406                        BEQ     7$              ;YES, RETURN
020720  010423                        MOV     R4,(R3)+        ;UPDATE ALL MODULE PASTIMES
```

```
3252  020722  010123                          MOV     R1,(R3)+        ;AND TIME EXTENDED BITS
3253  020724  000773                          BR      5$              ;CONTINUE
3254  020726  062703  000004          6$:     ADD     #4,R3           ;POINT TO NEXT PASS TIME ENTRY
3255  020732  000701                          BR      3$              ;GO DO IT AGAIN
3256  020734  004767  174012          7$:     JSR     PC,RST04        ;GO RESTORE R0 - R4
3257  020740  105767  160051                  TSTB    MODCNT          ;ANY MODULES STILL RUNNING ??
3258  020744  001002                          BNE     8$              ;BR IF YES
3259  020746  000167  166076                  JMP     CTRLCB          ;GO TERMINATE THIS "RUN"
3260  020752  000207                  8$:     RTS     PC              ;RETURN TO CALLING ROUTINE
```



```
3261                                  ;*************************************************************************
3262                                  ;THIS ROUTINE IS CALLED FROM THE "CKHUNG" ROUTINE AND IT IS USED
3263                                  ;TO DROP THE HUNG MODULE AND REPORT THE ERROR ON THE CONSOLE
3264                                  ;TTY. IF THE "TIMER" MODULE IS FOUND TO BE HUNG THEN THE ROUTINE
3265                                  ;DISABLES ANY FURTHER TIMING CHECKS USING THE CLOCK
3266                                  ;BY CLEARING THE BYTE TAGGED "CLOCK"
3267
3268  020754  052761  020000  000020  DRPHNG: BIS     #BIT13,STAT(R1) ;SET THE DROPPED BIT
3269  020762  032761  010000  000020          BIT     #BIT12,STAT(R1) ;IOMODX MODULE ??
3270  020770  001402                          BEQ     1$              ;BR IF NOT
3271  020772  105367  160024                  DECB    MDXCNT          ;DECREMENT COUNT OF IOMODX MODULES
3272  020776  105367  160013          1$:     DECB    MODCNT          ;DECREMENT COUNT OF RUNNING MODULES
3273  021002  021201                          CMP     (R2),R1         ;DID WE STOP THE TIMER MODULE ??
3274  021004  001014                          BNE     2$              ;BR IF NOT
3275  021006  105067  160341                  CLRB    CLOCK           ;DISABLE ALL TIMING CHECKS
3276  021012  010046                          MOV     R0,-(SP)                ;TYPEN SCREWS YOU IF YOU DON'T SAVE R0
3277  021014  004567  166330                  JSR     R5,TYPEN        ;GO REPORT DROPPING THE TIMER MODULE
3278  021020  022744                          CRCR
3279  021022  022744                          CRCR
3280  021024  024230                          TIMX1
3281  021026  022744                          CRCR
3282  021030  000000                          0
3283  021032  012600                          MOV     (SP)+,R0                ;RESTORE R0
3284  021034  000207                          RTS     PC              ;RETURN TO "CKHUNG"
3285
3286  021036  021267  160226          2$:     CMP     (R2),BKPTR      ;DID WE DROP AN ACTIVE BKMOD ?
3287  021042  001002                          BNE     3$              ;BR IF NOT
3288  021044  105067  157743                  CLRB    BRAKE           ;CLEAR THE BRAKE
3289  021050  004567  172560          3$:     JSR     R5,FILLNM       ;GO GET MOD NAME SET UP
3290  021054  026026                          TIMX+1                  ;POINTER TO ERROR MESSAGE
3291  021056  004767  176770                  JSR     PC,ERRTYM       ;GO SETUP TIME INFORMATION
3292  021062  010046                          MOV     R0,-(SP)                ;TYPEN SCREWS YOU IF YOU DON'T SAVE R0
3293  021064  004567  166260                  JSR     R5,TYPEN        ;REPORT THAT WE DROPPED IT
3294  021070  022744                          CRCR
3295  021072  022744                          CRCR
3296  021074  026025                          TIMX
3297  021076  026066                          TOTIM
3298  021100  022744                          CRCR
3299  021102  000000                          0
3300  021104  012600                          MOV     (SP)+,R0                ;RESTORE R0
3301  021106  000207                          RTS     PC              ;RETURN TO "CKHUNG" ROUTINE
3302
```

```
3303  021110  016767  160056  160050  LPON:   MOV     LPS,TPS
3304  021116  016767  160052  160044          MOV     LPB,TPB
3305  021124  012737  001526  000200          MOV     #TTYSRV,@#200
3306  021132  012737  000200  000202          MOV     #PRTY4,@#202
3307  021140  000167  167512                  JMP     COMCO3
3308  021144  177564                  LPWK1:  177564
3309  021146  177566                  LPWK2:  177566
3310                                  ;GET WRITE BUFFER ADDRESS SERVICE
3311
3312  021150  010146                  GWBUF,: MOV     R1,-(SP)        ;SAVE REGISTER
3313  021152  010246                          MOV     R2,-(SP)
3314  021154  017601  000004                  MOV     @4(SP),R1       ;GET ADDRESS OF MODULE
3315  021160  010102                          MOV     R1,R2
3316  021162  062701  000074                  ADD     #WBUFPA,R1
3317  021166  016721  160040                  MOV     WBUF,(R1)+      ;LOAD ADDRESS OF WRITE BUFFER
3318  021172  016721  160036                  MOV     WBFEA,(R1)+     ;LOAD EXTENDED ADDRESS BITS
3319  021176  012111                          MOV     (R1)+,(R1)      ;MOVE SIZE REQUESTED INTO SIZE ALLOWED
3320  021200  021167  ....J2                  CMP     (R1),MXWBF      ;REQUESTED SIZE OK?
3321  021204  101402                          BLOS    1$              ;YES- BRANCH
3322  021206  016711  160044                  MOV     MXWBF,(R1)      ;NO- SET ALLOWED SIZE
3323  021212  105767  157603          1$:     TSTB    MDXCTR          ;IF NOT YET SETUP, SKIP
3324  021216  100410                          BMI     2$
3325  021220  126267  000005  157575          CMPB    XFLAG(2),WBFLG  ;IF SAME, HAS ALREADY USED THIS BANK
3326  021226  001404                          BEQ     2$
3327  021230  105162  000005                  COMB    XFLAG(2)        ;IF NOT, SET FLAG TO SAME
3328  021234  105367  157561                  DECB    MDXCTR          ;AND COUNT DOWN MODULES
3329  021240  012602                  2$:     MOV     (SP)+,R2        ;RESTORE REGISTERS
3330  021242  012601                          MOV     (SP)+,R1
3331  021244  062716  000002                  ADD     #2,(SP)         ;SETUP RETURN PC
3332  021250  105367  157561                  DECB    ROT
3333  021254  100005                          BPL     3$
3334  021256  116767  160055  157551          MOVB    ROTN,ROT
3335  021264  004767  175014                  JSR     PC,ROTBUF       ;GO CHANGE WRITE BUFFER ADDRESS
3336  021270  000002                  3$:     RTI                     ;RETURN
3337
3338
```



```
3339                                  ;ROUTINE TO GET 18 BIT PHYSICAL ADDRESS OF VIRTUAL ADDRESS GIVEN,
3340                                  ;TOP 2 BITS ARE ALLIGNED TO POSITIONS 4 AND 5
3341                                  ;ASSUMES KERNEL I-SPACE ADDRESS IN CURRENT MAP
3342
3343  021272  010146                  GETPA,: MOV     R1,-(SP)        ;SAVE REGISTERS
3344  021274  010246                          MOV     R2,-(SP)
3345  021276  017601  000004                  MOV     @4(SP),R1       ;GET TABLE ADDRESS
3346  021302  012111                          MOV     (R1)+,(R1)      ;VA INTO PA LOC IN TABLE
3347  021304  105767  157774                  TSTB    KTPRES          ;KT11 IN USE?
3348  021310  001003                          BNE     1$              ;YES- BRANCH
3349  021312  005061  000002                  CLR     2(R1)           ;NO- CLEAR EA BITS
3350  021316  000436                          BR      3$              ;AND EXIT
3351
3352  021320  000311                  1$:     SWAB    (R1)
3353  021322  006211                          ASR     (R1)
3354  021324  006211                          ASR     (R1)
3355  021326  006211                          ASR     (R1)
3356  021330  006211                          ASR     (R1)
3357  021332  042711  177761                  BIC     #177761,(R1)    ;PAGE BITS ALLIGNED
3358  021336  062711  172340                  ADD     #KIPAR0,(R1)    ;KIPAR ADDRESS FORMED
3359  021342  017111  000000                  MOV     @(R1),(R1)      ;GET CONTENTS
3360  021346  012702  000006                  MOV     #6,R2           ;SHIFT 6 LEFT
3361  021352  006311                  2$:     ASL     (R1)
3362  021354  006161  000002                  ROL     2(R1)
3363  021360  005302                          DEC     R2
3364  021362  001373                          BNE     2$
3365  021364  014102                          MOV     -(R1),R2        ;GET VA
3366  021366  042702  160000                  BIC     #160000,R2      ;CLEAR PAGE BITS
3367  021372  005721                          TST     (P1)+
3368  021374  060221                          ADD     R2,(R1)+        ;CREATE 16 BIT PA
3369  021376  005511                          ADC     (R1)
3370  021400  042711  177774                  BIC     #177774,(R1)    ;ALLIGN BITS IN POSITIONS 4+5
3371  021404  006311                          ASL     (R1)
3372  021406  006311                          ASL     (P1)
3373  021410  006311                          ASL     (R1)
3374  021412  006311                          ASL     (R1)
3375  021414  012602                  3$:     MOV     (SP)+,R2        ;RESTORE REGISTERS
3376  021416  012601                          MOV     (SP)+,R1
3377  021420  062716  000002                  ADD     #2,(SP)
3378  021424  000002                          RTI                     ;RETURN
3379
3380
```

```
                                ;THIS ROUTINE ALLOWS RH70 DEVICES TO BE RUN WITH THE STANDARD MONITOR.


021426  010146                MAP22,: MOV     R1,-(SP)        ;SAVE REG. 1
021430  017601  000002                MOV     @2(SP),R1       ;GET THE 1ST ADDRESS OF THE TABLE
021434  012161  000002                MOV     (R1)+,2(R1)     ;GIVE BACK THE SAME 18 BITS
021440  011161  000004                MOV     (R1),4(R1)      ;THAT WERE GIVEN TO BE CONVERTED
021444  006261  000004                ASR     4(R1)           ;SHIFT BACK FROM POSITIONS 4 AND 5
021450  006261  000004                ASR     4(R1)           ;TO POSITIONS 0 AND 1
021454  006261  000004                ASR     4(R1)           ;
021460  006261  000004                ASR     4(R1)           ;
021464  012601                        MOV     (SP)+,R1        ;RESTORE REG. 1
021466  062716  000002                ADD     #2,(SP)         ;SKIP OVER VA TO GET THE RETURN PC
021472  000002                        RTI                     ;RETURN
```



```
                                ;MOVE CODE TO NEW OFFSET, AFTER CHECKING FOR CODE ALREADY IN
                                ;MIDST OF BEING MOVED. MAP KT11 TO NEW CODE AND RETURN TO IT.
                                ;ALL VALUES USED EXCEPT OFFSTD ARE ASSUMED CORRECT IN BANK 0 COPY
                                ;****  STACK POINTER IS REPOSITIONED TO SP%01 BEFORE RETURNING
                                ;MVCODS ENTRY POINT IS USED TO LEAVE OFFSET AS PREVIOUSLY SETUP-
                                ;NOTE, HOWEVER, THAT MAPRET MUST BE LOADED WITH THE RETURN ADDRESS
                                ;BEFORE CALLING. MVCODS SHOULD NOT BE CALLED UNLESS KTPRES IS SET
                                ;NOTE THAT MVCODE DISABLES THE TTY KEYBOARD

021474  105767  157604          MVCODE: TSTB    KTPRES          ;KT11 IN USE?
021500  001001                        BNE     .+4             ;YES- BRANCH
021502  000207                        RTS     PC              ;NO- RETURN (MOVING NOT ALLOWED W/O KT11)
021504  016700  157712                MOV     CLKHLD,R0       ;SAVE CONTENTS OF CLOCK CSR
021510  016705  157704                MOV     CLKADR,R5       ;SAVE ADR OF CLOCK CSR
021514  016703  157524                MOV     OFFSTD,R3       ;SAVE DESTINATION OFFSET
021520  011604                        MOV     (SP),R4         ;SAVE RETURN PC
021522  005037  177572                CLR     @#SSR0          ;TURN OFF KT11
021526  004767  170416                JSR     PC,KTSET0       ;MAP DOWN TO BANK 0
021532  052737  000001  177572        BIS     #BIT0,@#SSR0    ;TURN ON KT11
021540  010067  157656                MOV     R0,CLKHLD       ;STORE CONTENTS OF CLOCK CSR IN BANK 0
021544  010567  157650                MOV     R5,CLKADR       ;STORE ADR OF CSR IN BANK 0
021550  010367  157470                MOV     R3,OFFSTD       ;SET UP DESTINATION OFFSET IN BANK 0
                                                                ;COPY OF POINTER
021554  010467  157472                MOV     R4,MAPRET       ;SETUP RETURN PC
021560  105067  157561                CLRB    MOVFLG          ;CLEAR FLAG FOR 'ROTATED' MESSAGE
021564  005077  157372          MVCODS: CLR     @TKS            ;SHUT OUT TTY
021570  004767  173172                JSR     PC,CLRQUS
021574  105067  157230          MVCOD1: CLRB    XFRFLG
021600  012737  077406  172304        MOV     #77406,@#KIPDR2
021606  012737  077406  172306        MOV     #77406,@#KIPDR3
021614  105767  157505                TSTB    MOVING          ;CODE PARTIALLY RELOCATED?
021620  001406                        BEQ     1$              ;NO- BRANCH
021622  100067                        BPL     4$              ;IF PARTIALLY RELOCATED, BRANCH TO
                                                                ;FINISH RELOCATION
021624  016767  157416  157410        MOV     OFFSTM, OFFSET  ;IF FLAG NEGATIVE, INDICATES MOVE IS
021632  105067  157467                CLRB    MOVING          ;DONE BUT POINTERS NOT YET RESTORED
021636  026767  157402  157376  1$:   CMP     OFFSTD,OFFSET
021644  001004                        BNE     11$
021646  105167  157473                COMB    MOVFLG          ;SET FLAG NOT TO PRINT MESSAGE
021652  000167  000534                JMP     MVMAP
021656  016767  157362  157362  11$:  MOV     OFFSTD, OFFSTM  ;PUT DESTINATION OFFSET INTO SAVE LOCATION
021664  026767  157356  157350        CMP     OFFSTM, OFFSET  ;TO PREVENT LOSING VALUE IN MIDST OF MOVING
021672  101012                        BHI     2$              ;IF MOVING CODE UP, BRANCH
021674  016737  157342  172344        MOV     OFFSET,@#KIPAR2 ;IF MOVING DOWN, SETUP TO
021702  016737  157340  172346        MOV     OFFSTM,@#KIPAR3 ;DO LOWEST BANK FIRST
021710  112767  000001  157412        MOVB    #1,MVDOWN
021716  000450                        BR      MVCOD2
021720  016737  157300  172344  2$:   MOV     LCOR12,@#KIPAR2 ;IF MOVING UP, DO HIGHEST BANK FIRST
021726  032737  000177  172344        BIT     #177,@#KIPAR2
021734  001003                        BNE     3$
021736  162737  000200  172344        SUB     #200,@#KIPAR2
021744  042737  000177  172344  3$:   BIC     #177,@#KIPAR2   ;IF 4K BOUNDARY, DROP TO START OF LAST BANK OF CODE
021752  013737  172344  172346        MOV     @#KIPAR2,@#KIPAR3
021760  066737  157256  172344        ADD     OFFSET, @#KIPAR2
021766  066737  157254  172346        ADD     OFFSTM,@#KIPAR3
021774  105067  157330                CLRB    MVDOWN
```

```
3451  022000  000417                          BR      MVCOD2          ;GO TO ROUTINE THAT ACTUALLY MOVES THE CODE
3452  022002  112767  000001  157020  48:     MOVB    #1,XFRFLG       ;INDICATE ANOTHER PASS MUST BE MADE
3453  022010  016737  157234  172344          MOV     MOVBNK,@#KIPAR2 ;TO DO DESIRED MOVE AFTER CLEANING
3454  022016  016737  157226  172346          MOV     MOVBNK,@#KIPAR3 ;UP LAST ONE
3455  022024  166737  157212  172346          SUB     OFFSET,@#KIPAR3
3456  022032  066737  157210  172346          ADD     OFFSTM,@#KIPAR3
3457  022040  105067  157265          MVCOD2: CLRB    RETRY
3458  022044  013767  172344  157176          MOV     @#KIPAR2,MOVBNK ;SAVE CURRENT BANK BEING MOVED
3459  022052  112767  000001  157245          MOVB    #1,MOVING       ;INDICATE MOVE IN PROCESS
3460  022060  012701  040000          MVRTRY: MOV     #40000,R1
3461  022064  012702  060000                  MOV     #60000,R2
3462  022070  005737  172346                  TST     @#KIPAR3
3463  022074  001013                          BNE     2$
3464  022076  012122                  1$:     MOV     (R1)+,(R2)+     ;COPY VECTOR AREA DOWN TO BANK 0
3465  022100  032702  001000                  BIT     #1000,R2
3466  022104  001774                          BEQ     1$
3467  022106  013737  041270  061270          MOV     @#BKPTR+40000,@#BKPTR+60000     ;COPY BKPTR TO 0
3468  022114  012701  055000                  MOV     #55000,R1
3469  022120  012702  075000                  MOV     #75000,R2
3470  022124  012122                  2$:     MOV     (R1)+,(R2)+     ;MOVE CODE
3471  022126  032701  017777                  BIT     #17777,R1
3472  022132  001374                          BNE     2$
3473  022134  012701  040000                  MOV     #40000,R1
3474  022140  012702  060000                  MOV     #60000,R2
3475  022144  005737  172346                  TST     @#KIPAR3
3476  022150  001014                          BNE     5$
3477  022152  021112                  3$:     CMP     (R1),(R2)
3478  022154  001402                          BEQ     4$
3479  022156  000167  000402                  JMP     MVERR
3480  022162  022122                  4$:     CMP     (R1)+,(R2)+
3481  022164  032702  001000                  BIT     #1000,R2
3482  022170  001770                          BEQ     3$
3483  022172  012701  055000                  MOV     #55000,R1
3484  022176  012702  075000                  MOV     #75000,R2
3485  022202  021112                  5$:     CMP     (R1),(R2)       ;NEW COPY OK?
3486  022204  001402                          BEQ     6$              ;YES, CONTINUE
3487  022206  000167  000352                  JMP     MVERR           ;NO- BRANCH TO REPORT ERROR
3488  022212  022122                  6$:     CMP     (R1)+,(R2)+
3489  022214  032701  017777                  BIT     #17777,R1
3490  022220  001370                          BNE     5$
3491  022222  026767  157020  157012          CMP     OFFSTM,OFFSET
3492  022230  101413                          BLOS    7$
3493  022232  162737  000200  172344          SUB     #200,@#KIPAR2
3494  022240  162737  000200  172346          SUB     #200,@#KIPAR3
3495  022246  023767  172346  156772          CMP     @#KIPAR3,OFFSTM
3496  022254  103271                          BHIS    MVCOD2
3497  022256  000415                          BR      MVCOD3
3498  022260  062737  000200  172344  7$:     ADD     #200,@#KIPAR2
3499  022266  062737  000200  172346          ADD     #200,@#KIPAR3
3500  022274  016701  156746                  MOV     OFFSTM,R1
3501  022300  066701  156720                  ADD     LCOR12,R1
3502  022304  023701  172346                  CMP     @#KIPAR3,R1
3503  022310  103653                          BLO     MVCOD2
3504  022312  112767  177777  157005  MVCOD3: MOVB    #-1,MOVING      ;INDICATE MOVE DONE BUT POINTERS
3505  022320  016767  156722  156714          MOV     OFFSTM,OFFSET   ;NOT YET RESTORED
3506  022326  016737  156710  172344          MOV     OFFSET,@#KIPAR2
```



```
3507  022334  012701  001242                  MOV     #OFFSET,R1
3508  022340  062701  040000                  ADD     #40000,R1
3509  022344  016711  156672                  MOV     OFFSET,(R1)     ;SET OFFSET VALUE INTO NEW COPY
3510  022350  105067  156751                  CLRB    MOVING
3511  022354  105767  156750                  TSTB    MVDOWN
3512  022360  001402                          BEQ     MVMSG
3513  022362  004767  002304                  JSR     PC,LOADWC       ;RELOAD WORST CASE FROM LCOR12+OFFSET
3514                                                                  ;TO HCOR12
3515  022366  005067  156466          MVMSG:  CLR     IOBKID
3516  022372  112767  000001  156413          MOVB    #1,BRAKE
3517  022400  105767  156424                  TSTB    XFRFLG          ;FINAL MOVE NOT YET DONE?
3518  022404  001402                          BEQ     MVMAP
3519  022406  000167  177162                  JMP     MVCOD1          ;YES- DO DESIRED MOVE NOW THAT PREVIOUS ONE
3520                                                                  ;HAS BEEN CLEANED UP
3521  022412  016704  156634          MVMAP:  MOV     MAPRET,R4       ;GET RETURN PC
3522  022416  016701  156620                  MOV     OFFSET,R1       ;GET NEW OFFSET
3523  022422  062701  000200                  ADD     #200,R1
3524  022426  012702  172302                  MOV     #KIPDR1,R2
3525  022432  012703  000006                  MOV     #6,R3
3526  022436  012712  077406          1$:     MOV     #77406,(R2)     ;SET PDR TO 4K,RW,UP
3527  022442  010162  000040                  MOV     R1,40(R2)       ;MAP PAR TO DESIRED 4K
3528  022446  062702  000002                  ADD     #2,R2           ;MOVE POINTER TO NEXT PAGE
3529  022452  062701  000200                  ADD     #200,R1         ;CHANGE TO NEXT 4K VALUE
3530  022456  005303                          DEC     R3
3531  022460  001366                          BNE     1$
3532  022462  116701  156657                  MOVB    MOVFLG,R1       ;SAVE THE FLAG
3533  022466  016737  156550  172340          MOV     OFFSET,@#KIPAR0 ;CHANGE TO NEW CODE
3534  022474  012706  027212                  MOV     #SPBOT,SP
3535  022500  005037  177776                  CLR     @#PSW
3536  022504  105701                          TSTB    R1              ;OK TO PRINT MESSAGE ?
3537  022506  100425                          BMI     2$              ;NO, CONTINUE
3538  022510  032777  001000  156460          BIT     #BIT9,@SR       ;YES, BUT IS IT INHIBITED ?
3539  022516  001021                          BNE     2$              ;YES, CONTINUE
3540  022520  104406  024446                  MSG$,RELMSG             ;"CODE RELOCATED TO"
3541  022524  004567  172142                  JSR     R5,TO18S        ;GET 18 BIT ADDRESS
3542  022530  001242                          OFFSET
3543  022532  001206                          MEMPA
3544  022534  001210                          MEMEA
3545  022536  016746  156444                  MOV     MEMPA,-(SP)
3546  022542  016746  156442                  MOV     MEMEA,-(SP)
3547  022546  004767  170564                  JSR     PC,ITOAX        ;TYPE ADDRESS
3548  022552  012746  022772                  MOV     #ACRLF,-(SP)
3549  022556  004767  164604                  JSR     PC,TYPE
3550  022562  010407                  2$:     MOV     R4,PC           ;RETURN FROM MVCODE CALL
3551
3552
3553  022564  105067  156270          MVERR:  CLRB    IOBKID
3554  022570  112767  000001  156215          MOVB    #1,BRAKE
3555  022576  010267  156402                  MOV     R2,MEMVA
3556  022602  104406  024420                  MSG$,RELERR             ;"RELOCATION ERROR AT"
3557  022606  104413  001204                  GETPA$,MEMVA
3558  022612  016746  156370                  MOV     MEMPA,-(SP)
3559  022616  016746  156366                  MOV     MEMEA,-(SP)
3560  022622  004767  170510                  JSR     PC,ITOAX        ;TYPE ADDRESS
3561  022626  012746  022772                  MOV     #ACRLF,-(SP)
3562  022632  004767  164530                  JSR     PC,TYPE
```

```
3563  022636  105267  156467                  INCB    RETRY
3564  022642  126727  156463  000003          CMPB    RETRY,#3
3565  022650  002002                          BGE     6$
3566  022652  000167  177202                  JMP     MVRTRY
3567  022656  104406  024100          6$:     MSG$,FATAL              ;"UNABLE TO RELOCATE- FATAL, RELOAD."
3568  022662  104406  023746                  MSG$,HLT                ;"HALT"
3569  022666  000000                  7$:     HALT
3570  022670  000776                          BR      7$
```



```
3571                                    ;PURE MESSAGES
3572
3573  022672  042045  041505  054057  TITLE:  .ASCIZ  '%DEC/X11,  DXQAB-M  STANDARD MONITOR'
3574  022700  030461  020056  042040
3575  022706  050530  041101  046455
3576  022714  020040  052123  047101
3577  022722  040504  042122  046440
3578  022730  047117  052111  051117
3579  022736      000
3580  022737      045  000056          DOT:    .ASCIZ  '%.'
3581  022742  000045                  CR:     .ASCIZ  '%'
3582  022744  022445      000         CRCR:   .ASCIZ  '%%'
3583  022747      040  052101      000 AAT:    .ASCIZ  ' AT'
3584  022753      045  052522  020116  SUMARY: .ASCIZ  '%RUN SUMMARY'
3585  022760  052523  046515  051101
3586  022766  000131
3587          022772                  $CRLF=.+2
3588  022770  041536  000045          CTRLC:  .ASCIZ  '^C%'
3589  022774  022445  050440  042525  QOVRFL: .ASCIZ  '%% QUEUE OVERFLOW --- SYSTEM CRASH%'
3590  023002  042525  047440  042526
3591  023010  043122  047514  020127
3592  023016  026455  020055  054523
3593  023024  052123  046505  041440
3594  023032  040522  044123  000045
3595  023040  022445  044440  053116  INVTRP: .ASCIZ  '%% INVALID TRAP CALL --- SYSTEM CRASH%'
3596  023046  046101  042111  052040
3597  023054  040522  020120  040503
3598  023062  046114  026440  026455
3599  023070  051440  051531  042524
3600  023076  020115  051103  051501
3601  023104  022510      000
3602  023107      045  047111  040526  INVCMD: .ASCIZ  '%INVALID COMMAND%'
3603  023114  044514  020104  047503
3604  023122  046515  047101  022504
3605  023130      000
3606  023131      045  052522  020116  INVCD1: .ASCIZ  '%RUN MODE --- ONLY A CNTL C IS PERMITTED%'
3607  023136  047515  042504  026440
3608  023144  026455  047440  046116
3609  023152  020131  020101  047103
3610  023160  046124  041440  044440
3611  023166  020123  042520  046522
3612  023174  052111  042524  022504
3613  023202      000
3614  023203      045  051120  043517  INVCD2: .ASCIZ  '%PROGRAM CANNOT GO THAT HIGH%'
3615  023210  040522  020115  040503
3616  023216  047116  052117  043440
3617  023224  020117  044124  052101
3618  023232  044040  043511  022510
3619  023240      000
3620  023241      045  047516  053440  INVCD3: .ASCIZ  '%NO WAY!   MEMORY MANAGEMENT MUST BE IN USE%'
3621  023246  054501  020041  020040
3622  023254  042515  047515  054522
3623  023262  046440  047101  043501
3624  023270  046505  047105  020124
3625  023276  052515  052123  041040
3626  023304  020105  047111  052440
```

```
3627  023312  042523  000045
3628  023316  047045  020117  047515  INVCD4: .ASCIZ  '%NO MODULES SELECTED%'
3629  023324  052504  042514  020123
3630  023332  042523  042514  052103
3631  023340  042105  000045
3632  023344  044445  053116  046101  INVADR: .ASCIZ  '%INVALID ADDR/DATA%'
3633  023352  042111  040440  042104
3634  023360  027522  040504  040524
3635  023366  000045
3636  023370  046445  042117  046125  INVNAM: .ASCIZ  '%MODULE NAME NOT FOUND%'
3637  023376  020105  040516  042515
3638  023404  047040  052117  043040
3639  023412  052517  042116  000045
3640  023420  020045  020040  045050  SYSER1: .ASCIZ  '%   (JAM,SRV,PBA,CUA,FLG/INT,WHAMI,CDATA,CTAG/CPU)% '
3641  023426  046501  051454  053122
3642  023434  050054  040502  041454
3643  023442  040525  043034  043514
3644  023450  044457  052116  053454
3645  023456  040510  044515  041454
3646  023464  040504  040524  041454
3647  023472  040524  027507  050103
3648  023500  024525  020045     000
3649  023505     045  041113  043125  KBOFLO: .ASCIZ  '%KBUF OFLO'
3650  023512  047440  046106  000117
3651  023520  022445  047524  042440  CLR40:  .ASCIZ  '%%TO EXERCISE LOAD MEDIUM YOU MUST CLEAR LOC 40%'
3652  023526  042530  041522  051511
3653  023534  020105  047514  042101
3654  023542  046440  042105  052511
3655  023550  020115  047531  020125
3656  023556  052515  052123  041440
3657  023564  042514  051101  046040
3658  023572  041517  032040  022460
3659  023600     000
3660  023601     045  054523  020123  SYSERR: .ASCIZ  '%SYS ERR (VIRT,PC,PSW,SP,VECT):
3661  023606  051105  020122  053050
3662  023614  051111  027124  041520
3663  023622  050054  053523  051454
3664  023630  026120  042526  052103
3665  023636  035051  020040     000
3666  023643     045  040503  044103  CONN:   .ASCIZ  '%CACHE ENABLED'
3667  023650  020105  047105  041101
3668  023656  042514  000104
3669  023662  041445  041501  042510  COFFF:  .ASCIZ  '%CACHE DISABLED'
3670  023670  042040  051511  041101
3671  023676  042514  000104
3672  023702  022445  047520  042527  PWRFAI: .ASCIZ  '%%POWER FAILURE%%'
3673  023710  020122  040506  046111
3674  023716  051125  022505  000045
3675  023724  051045  047125  046440  MEMCHG: .ASCIZ  '%RUN MEMORY TESTS'
3676  023732  046505  051117  020131
3677  023740  042524  052123  000123
3678  023746  044045  046101  022524  HLT:    .ASCIZ  '%HALT%%'
3679  023754  000045
3680  023756  050045  051101  052111  PENABL: .ASCIZ  '%PARITY ENABLED'
3681  023764  020131  047105  041101
3682  023772  042514  000104
```



```
3683  023776  047045  020117  052113  NOKT:   .ASCIZ  '%NO KT11'
3684  024004  030461     000
3685  024007     045  052113  030461  KTDIS:  .ASCIZ  '%KT11 OFF'
3686  024014  047440  043106     000
3687  024021     045  047516  050040  NOPAR:  .ASCIZ  '%NO PARITY'
3688  024026  051101  052111  000131
3689  024034  050045  051101  052111  PAOFF:  .ASCIZ  '%PARITY OFF'
3690  024042  020131  043117  000106
3691  024050  022445  052113  030461  KTFATL: .ASCIZ  '%%KT11 ABORT --- FATAL%'
3692  024056  040440  047502  052122
3693  024064  026440  026455  043040
3694  024072  052101  046101  000045
3695  024100  052445  040516  046102  FATAL:  .ASCIZ  '%UNABLE TO RELOCATE- FATAL, RELOAD %'
3696  024106  020105  047524  051040
3697  024114  046105  041517  052101
3698  024122  026505  043040  052101
3699  024130  046101  020056  042522
3700  024136  047514  042101  022440
3701  024144     000
3702  024145     045  022445  025045  SYSBAD: .ASCIZ  '%%%%*** TOO MANY SYS ERRORS --- EXERCISE ABORTED%%'
3703  024152  025052  052040  047517
3704  024160  046440  047101  020131
3705  024166  054523  020123  051105
3706  024174  047522  051522  026440
3707  024202  026455  042440  042530
3708  024210  041522  051511  020105
3709  024216  041101  051117  042524
3710  024224  022504  000045
3711  024230  041445  047514  045503  TIMX1:  .ASCIZ  '%CLOCK MODULE DROPPED, ALL TIMING FUNCTIONS DISABLED.%'
3712  024236  046440  042117  046125
3713  024244  020105  051104  050117
3714  024252  042520  027104  040440
3715  024260  046114  052040  046511
3716  024266  047111  020107  052506
3717  024274  041516  044524  047117
3718  024302  020123  044504  040523
3719  024310  046102  042105  022456
3720  024316     000
```

```
3721  024317  045  047045  020117  NOCLK:  .ASCIZ  '%%NO SYSTEM CLOCK---RESTART @ 1000 OR HIT CONTINUE FOR NO CLOCK%'
3722  024324  054523  052123  046505
3723  024332  041440  047514  045503
3724  024340  026455  051055  051505
3725  024346  040524  052122  040040
3726  024354  030440  030060  020060
3727  024362  051117  044040  052111
3728  024370  041440  047117  044524
3729  024376  052516  020105  047506
3730  024404  020122  047516  041440
3731  024412  047514  045503  000045
3732  024420  051045  046105  041517  RELERR:  .ASCIZ  '%RELOCATION ERROR AT '
3733  024426  052101  047511  020116
3734  024434  051105  047522  020122
3735  024442  052101  000040
3736  024446  051045  046105  041517  RELMSG:  .ASCIZ  '%RELOCATED TO '
3737  024454  052101  042105  052040
3738  024462  020117  000
3739  024465  045  054523  052123  SYSSIZ:  .ASCIZ  '%SYSTEM SIZE:'
3740  024472  046505  051440  055111
3741  024500  035105  000
3742  024503  113  000  KAY:  .ASCIZ  'K'
3743  024505  045  051127  052111  ROTENB:  .ASCIZ  '%WRITE BUFFER ROTATION ENABLED, RANGE: '
3744  024512  020105  052502  043106
3745  024520  051105  051040  052117
3746  024526  052101  047511  020116
3747  024534  047105  041101  042514
3748  024542  027104  051040  047101
3749  024550  042507  020072  000
3750
3751  024555  045  051127  052111  ROTDIS:  .ASCIZ  '%WRITE BUFFER ROTATION DISABLED'
3752  024562  020105  052502  043106
3753  024570  051105  051040  052117
3754  024576  052101  047511  020116
3755  024604  044504  040523  046102
3756  024612  042105  000
3757  024615  045  052113  030461  KTONM:  .ASCIZ  '%KT11 ENABLED'
3758  024622  042440  040516  046102
3759  024630  042105  000
3760  024633  045  053523  052111  NOSWR:  .ASCIZ  '%SWITCH REGISTER AT LOC. 1004%'
3761  024640  044103  051040  043505
3762  024646  051511  042524  020122
3763  024654  052101  046040  041517
3764  024662  020056  030061  032060
3765  024670  000045
3766                                  .EVEN
```



```
3767                          ;ROUTINE TO LOAD WORST CASE UNIBUS PATTERN IN ALL FREE MEMORY
3768
3769  024672  004467  170042  LOADWC: JSR     R4,SAV04
3770  024676  016702  154322          MOV     LCOR12,R2       ;SETUP 12 BIT STARTING ADDRESS
3771  024702  062702  154334          ADD     OFFSET,R2
3772  024706  026702  154314          CMP     HCOR12,R2       ;ANY MEMORY TO LOAD ?
3773  024712  001425                  BEQ     8$              ;NO, GET OUT
3774  024714  016700  154300          MOV     LOCORE,R0       ;IF NO KT11, R0 WILL CONTAIN PHYSICAL ADDRESS
3775  024720  012704  025006  2$:     MOV     #WCASE,R4       ;LOAD ADDRESS OF WORST CASE DATA TABLE
3776  024724  105767  154354  3$:     TSTB    KTPRES          ;USE KT11?
3777  024730  001404                  BEQ     5$              ;NO- BRANCH
3778  024732  012700  040000          MOV     #40000,R0       ;SET VIRTUAL ADDRESS TO THE START OF PAGE 2
3779  024736  010267  145402          MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL ADDRESS
3780  024742  012703  000400  5$:     MOV     #256.,R3        ;SET UP COUNTER
3781  024746  011420  6$:     MOV     (R4),(R0)+      ;LOAD DATA
3782  024750  077303                  DEC     R3
3783  024752  001375                  BNE     6$
3784  024754  062702  000010          ADD     #10,R2          ;AFTER EACH 256 WORDS, REMAP
3785  024760  020267  154242          CMP     R2,HCOR12       ;DONE?
3786  024764  002403                  BLT     7$              ;NO- BRANCH
3787  024766  004767  167760  8$:     JSR     PC,FST04
3788  024772  000207                  RTS     PC              ;YES- EXIT
3789  024774  005724  7$:     TST     (R4)+           ;MOVE DATA POINTER
3790  024776  020427  025106          CMP     R4,#WCASEE      ;TO NEXT WORST CASE VALUE
3791  025002  103760                  BLO     3$              ;AND CONTINUE
3792  025004  000746                  BR      2$
3793
3794  025006  177776  000001  177775  WCASE:  .WORD   177776,1,177775,2,177773,4,177767,10
3795  025014  000002  177773  000004
3796  025022  177767  000010
3797  025026  177757  000020  177737          .WORD   177757,20,177737,40,177677,100,177577,200
3798  025034  000040  177677  000100
3799  025042  177577  000200
3800  025046  177377  000400  176777          .WORD   177377,400,176777,1000,175777,2000,173777,4000
3801  025054  001000  175777  002000
3802  025062  173777  004000
3803  025066  167777  010000  157777          .WORD   167777,10000,157777,20000,137777,40000,77777,100000
3804  025074  020000  137777  040000
3805  025102  077777  100000
3806  025106                  WCASEE:
```

```
 3807                                   ;COMMAND TABLE. KEYBOARD COMMANDS ARE MATCHED AGAINST IT.
 3808                                   ;THE MATCH MUST BE EXACT. NO ABBREVIATIONS ARE ALLOWED.
 3809
 3810  025106                           COMTAB:
 3811  025106                                   TOKN    "RUNL",RUNL
 3812  025106  052522  046116     000           .ASCIZ  %RUNL%
 3813  025114  010700                           .WORD   RUNL
 3814  025116                                   TOKN    "RUN",RUN
 3815  025116  052522  000116                   .ASCIZ  %RUN%
 3816  025124  010710                           .WORD   RUN
 3817  025126                                   TOKN    "MOD",MOD
 3818  025126  047515  000104                   .ASCIZ  %MOD%
 3819  025134  013222                           .WORD   MOD
 3820  025136                                   TOKN    "SEL",SEL
 3821  025136  042523  000114                   .ASCIZ  %SEL%
 3822  025144  012462                           .WORD   SEL
 3823  025146                                   TOKN    "DES",DES
 3824  025146  042504  000123                   .ASCIZ  %DES%
 3825  025154  012500                           .WORD   DES
 3826  025156                                   TOKN    "MAP",MAP
 3827  025156  040515  000120                   .ASCIZ  %MAP%
 3828  025164  012314                           .WORD   MAP
 3829  025166                                   TOKN    "FILL",FILL
 3830  025166  044506  046114     000           .ASCIZ  %FILL%
 3831  025174  013214                           .WORD   FILL
 3832  025176                                   TOKN    "KTOFF",KTOFF
 3833  025176  052113  043117  000106           .ASCIZ  %KTOFF%
 3834  025206  012562                           .WORD   KTOFF
 3835  025210                                   TOKN    "KTON",KTON
 3836  025210  052113  047117     000           .ASCIZ  %KTON%
 3837  025216  012632                           .WORD   KTON
 3838  025220                                   TOKN    "PON",PON
 3839  025220  047520  000116                   .ASCIZ  %PON%
 3840  025226  013004                           .WORD   PON
 3841  025230                                   TOKN    "POFF",POFF
 3842  025230  047520  043106     000           .ASCIZ  %POFF%
 3843  025236  013044                           .WORD   POFF
 3844  025240                                   TOKN    "ROTON",ROTON
 3845  025240  047522  047524  000116           .ASCIZ  %ROTON%
 3846  025250  012744                           .WORD   ROTON
 3847  025252                                   TOKN    "ROTOFF",ROTOFF
 3848  025252  047522  047524  043106           .ASCIZ  %ROTOFF%
 3849  025260     000
 3850  025262  012760                           .WORD   ROTOFF
 3851  025264                                   TOKN    "LPON",LPON
 3852  025264  050114  047117     000           .ASCIZ  %LPON%
 3853  025272  021110                           .WORD   LPON
 3854  025274                                   TOKN    "CON",CON
 3855  025274  047503  000116                   .ASCIZ  %CON%
 3856  025302  013130                           .WORD   CON
 3857  025304                                   TOKN    "COFF",COFF
 3858  025304  047503  043106     000           .ASCIZ  %COFF%
 3859  025312  013170                           .WORD   COFF
 3860  025314     015     000                   .BYTE   15,0            ;CR
 3861  025316     004     000                   .BYTE   4,0
 3862  025320  010646                           .WORD   COMCO2
```



```
 3863  025322  000000  000000  000000           .WORD   0,0,0
```

```
                                        ;IMPURE MESSAGES.
                                        
025330  050045  051101  052111  PERR1:  .ASCII  '%PARITY ERROR   '
025336  020131  051105  047522
025344  020122  020040
025350  020040  020040  020040  PREG:   .ASCII  '      /'
025356     057
025357     040  020040  020040  PCONT:  .ASCII  '      '
025364     040
025365     040  020040  052522  RUNDIA: .ASCIZ  '   RUN PARITY/MEMORY DIAGNOSTICS%'
025372  020116  040520  044522
025400  054524  046457  046505
025406  051117  020131  044504
025414  043501  047516  052123
025422  041511  022523     000
025427     045  040520  044522  PERR2:  .ASCIZ  '%PARITY ERROR --- (JAM,SPV,PBA,CUA,FLG/INT,WHAMI,CDATA,CTAG/CPU)%'
025434  054524  042440  051122
025442  051117  026440  026455
025450  024040  040512  026115
025456  051123  026126  041120
025464  026101  052503  026101
025472  046106  027507  047111
025500  026124  034127  046501
025506  026111  042103  052101
025514  026101  052103  043501
025522  041457  052520  022451
025530     000
025531     045  054523  052123  NUMSYS: .ASCIZ  '%SYSTEM ERRORS:'
025536  046505  042440  051122
025544  051117  035123     000
025551     040  020040  020040  NUMPWR: .ASCIZ  '     POWER FAILS:'
025556  047520  042527  020122
025564  040506  046111  035123
025572     000
025573     045  020040  020040  AMODNM: .ASCII  '%      AT '
025600  020040  052101     040
025605     040  020040  020040  APC:    .ASCII  '       STAT '
025612  020040  052123  052101
025620     040
025621     040  020040  020040  AMDSTA: .ASCII  '      '
025626     040
025627     040  050040  051501  MDIRE:  .ASCII  '  PASCNT '
025634  047103  020124
025640  020040  020040  027040  APSCNT: .ASCII  '     .  ERRCNT '
025646  020040  051105  041522
025654  052116     040
025657     040  020040  020040  AERRS:  .ASCIZ  '     .'
025664  000056
                025667          .ODD
025667     040  020040  020040  HOLD5:  .ASCIZ  '     '
025674     000
025675     040  020040  020040  HOLD6:  .ASCIZ  '        '
025702  020040  000040
025706  020040  020040  020040  HOLD8:  .ASCIZ  '          '
025714  020040  020040     000
025721     040  020040  020040  AEND:   .ASCII  '       PC='
```



```
025726  020040  041520     055
025733     040  020040  020040  AEND1:  .ASCII  '       APC='
025740  020040  050101  026503
025746  020040  020040  020040  AEND2:  .ASCIZ  '       '
025754  000040
025756  020040  040520  051523  $PASS:  .ASCII  '  PASS# '
025764  020043
025766  020040  020040  027040  APASS:  .ASCIZ  '     .'
025774     000
025775     040  051104  050117  MODEND: .ASCIZ  ' DROPPED%'
026002  042520  022504     000
026007     105  042116  040520  APSEND: .ASCII  'ENDPAS='
026014  026523
026016  020040  020040  027040  BPSCNT: .ASCIZ  '     .'
026024     000
026025     045  020040  020040  TIMX:   .ASCIZ  '%      IS HUNG, MODULE DROPPED.%'
026032  020040  051511  044040
026040  047125  027107  046440
026046  042117  046125  020105
026054  051104  050117  042520
026062  027104  000045
026066  051040  047125  044524  TOTIM:  .ASCII  ' RUNTIME='
026074  042515     055
026077     040  020040     072  THRS:   .ASCII  '   :'
026103     040  035040          TMINS:  .ASCII  '  :'
026106  020040     000          TSECS:  .ASCIZ  '  '
026111     040  051520  044524  PSTIM:  .ASCII  ' PSTIME='
026116  042515     055
026121     040  035040          MMINS:  .ASCII  '  :'
026124  020040  000045          MSECS:  .ASCIZ  '  %'
026130  041445  051123  020101  AERROR: .ASCII  '%CSRA '
026136  020040  020040  020040  ACSRAC: .ASCII  '       CSRC '
026144  041440  051123  020103
026152  020040  020040  020040  ACSRC:  .ASCII  '       STATC '
026160  051440  040524  041524
026166     040
026167     040  020040  020040  ASTATC: .ASCIZ  '       '
026174  020040     000
026177     045  051503  040522  ADTERR: .ASCII  '%CSRA '
026204     040
026205     040  020040  020040  ADTE6:  .ASCII  '       S/B '
026212  020040  027523  020102
026220  020040  020040  020040  ADTE3:  .ASCII  '       WAS '
026226  053440  051501     040
026233     040  020040  020040  ADTE2:  .ASCII  '       WRADR '
026240  020040  051127  042101
026246  020122
026250  020040  020040  020040  ADTE5:  .ASCII  '       RDADR '
026256  051040  040504  051104
026264     040
026265     040  020040  020040  ADTE4:  .ASCIZ  '       '
026272  020040     000
026275     040  040504  040524  ADTE4A: .ASCIZ  ' DATA ERROR'
026302  042440  051122  051117
026310     000
026311     127  051117  021504  ADTEX:  .ASCII  'WORD# '
```

```
3976  026316  040
3977  026317  040  020040  020040  ADTE7:  .ASCIZ  '     .%'
3978  026324  022456  000
3979  026327  040  051105  021522  ERRNMB: .ASCII  ' ERR# '
3980  026334  040
3981  026335  040  020040  020040  AERNMB: .ASCIZ  '     .'
3982  026342  000056
3983  026344  046445  046505  043040  MERR1:  .ASCII  '%MEM FAILURE= PA '
3984  026352  044501  052514  042522
3985  026360  020055  040520  040
3986  026365  040  020040  020040  MERR2:  .ASCII  '       S/B '
3987  026372  020040  027523  020102
3988  026400  020040  020040  020040  MERR3:  .ASCII  '       WAS '
3989  026406  053440  051501  040
3990  026413  040  020040  020040  MERR4:  .ASCIZ  '       '
3991  026420  020040  000
3992  026423  045  020040  020040  AERNM:  .ASCII  '%      HAD '
3993  026430  020040  040520  020104
3994  026436  020040  020040  027040  AERCT:  .ASCII  '     . DATA ERRORS OUT OF '
3995  026444  042040  052101  020101
3996  026452  051105  047522  051522
3997  026460  047440  052125  047440
3998  026466  020106
3999  026470  020040  020040  027040  AERTOT: .ASCIZ  '     . WORDS READ%'
4000  026476  053440  051117  051504
4001  026504  051040  040505  022504
4002  026512  000
4003  026513  040  040  040  DECVAL: .BYTE   40,40,40,40,40,40
4004  026516  040  040  040
```



```
4005                                    ;BUFFER AREAS.
4006
4007          026522                    .EVEN
4009  026522                    MODQ:                       ;ROOM FOR 60 MODULE POINTERS.
4009  026712                    KBUF:                 ;KEYBOARD BUFFER, KBUFL LONG
4010  026752  000014                   .BLKW   14
4011  027002  000004            USP:    .BLKW   4
4012  027012                    SSP:
4013  027012  000100                   .BLKW   64.             ;MONITOR STACK AREA
4014  027212                    SPBOT:
4015  027212  000020                   .BLKW   16.             ;SPECIAL MONITOR STACK AREA
4016  027252                    SPSPEC:
4017  027252  000620            IOQ:    .BLKB   IOQL
4018  030072  001130            TYPEQ:  .BLKB   TYPQL
4019
4020  031222  000000                   OPEN                    ;PATCH AREA
4021  031224  000000                   OPEN                    ;PATCH AREA
4022  031226  000000                   OPEN                    ;PATCH AREA
4023  031230  000000                   OPEN                    ;PATCH AREA
4024  031232  000000                   OPEN                    ;PATCH AREA
4025  031234  000000                   OPEN                    ;PATCH AREA
4026  031236  000000                   OPEN                    ;PATCH AREA
4027  031240  000000                   OPEN                    ;PATCH AREA
4028  031242  000000                   OPEN                    ;PATCH AREA
4029  031244  000000                   OPEN                    ;PATCH AREA
4030  031246  000000                   OPEN                    ;PATCH AREA
4031  031250  000000                   OPEN                    ;PATCH AREA
4032  031252  000000                   OPEN                    ;PATCH AREA
4033  031254  000000                   OPEN                    ;PATCH AREA
4034  031256  000000                   OPEN                    ;PATCH AREA
4035  031260  000000                   OPEN                    ;PATCH AREA
4036  031262  000000                   OPEN                    ;PATCH AREA
4037  031264  000000                   OPEN                    ;PATCH AREA
4038  031266  000000                   OPEN                    ;PATCH AREA
4039  031270  000000                   OPEN                    ;PATCH AREA
4040  031272  000000                   OPEN                    ;PATCH AREA
4041  031274  000000                   OPEN                    ;PATCH AREA
4042  031276  000000                   OPEN                    ;PATCH AREA
4043  031300  000000                   OPEN                    ;PATCH AREA
4044  031302  000000                   OPEN                    ;PATCH AREA
4045  031304  000000                   OPEN                    ;PATCH AREA
4046  031306  000000                   OPEN                    ;PATCH AREA
4047  031310  000000                   OPEN                    ;PATCH AREA
4048  031312  000000                   OPEN                    ;PATCH AREA
4049  031314  000000                   OPEN                    ;PATCH AREA
4050  031316  000000                   OPEN                    ;PATCH AREA
4051  031320  000000                   OPEN                    ;PATCH AREA
4052  031322  000000                   OPEN                    ;PATCH AREA
4053  031324  000000                   OPEN                    ;PATCH AREA
4054  031326  000000                   OPEN                    ;PATCH AREA
4055  031330  000000                   OPEN                    ;PATCH AREA
4056  031332  000000                   OPEN                    ;PATCH AREA
4057  031334  000000                   OPEN                    ;PATCH AREA
4058  031336  000000                   OPEN                    ;PATCH AREA
4059  031340  000000                   OPEN                    ;PATCH AREA
4060  031342  000000                   OPEN                    ;PATCH AREA
```

```
 4061  031344  000000                        OPEN                    ;PATCH AREA
 4062  031346  000000                        OPEN                    ;PATCH AREA
 4063  031350  000000                        OPEN                    ;PATCH AREA
 4064  031352  000000                        OPEN                    ;PATCH AREA
 4065  031354  000000                        OPEN                    ;PATCH AREA
 4066  031356  000000                        OPEN                    ;PATCH AREA
 4067  031360  000000                        OPEN                    ;PATCH AREA
 4068  031362  000000                        OPEN                    ;PATCH AREA
 4069  031364  000000                        OPEN                    ;PATCH AREA
 4070  031366  000000                        OPEN                    ;PATCH AREA
 4071  031370  000000                        OPEN                    ;PATCH AREA
 4072  031372  000000                        OPEN                    ;PATCH AREA
 4073  031374  000000                        OPEN                    ;PATCH AREA
 4074  031376  000000                        OPEN                    ;PATCH AREA
 4075  031400  000000                        OPEN                    ;PATCH AREA
 4076  031402  000000                        OPEN                    ;PATCH AREA
 4077  031404  000000                        OPEN                    ;PATCH AREA
 4078  031406  000000                        OPEN                    ;PATCH AREA
 4079  031410  000000                        OPEN                    ;PATCH AREA
 4080  031412  000000                        OPEN                    ;PATCH AREA
 4081  031414  000000                        OPEN                    ;PATCH AREA
 4082  031416  000000                        OPEN                    ;PATCH AREA
 4083  031420  000000                        OPEN                    ;PATCH AREA
 4084
 4085  031422  000200              LOGBUF: .BLKW   200               ;BUFFER HOLDS LAST COPY TAKEN OF 11/60 LOG
 4086
 4087          032022                      .=.                     ;FIRST ADDRESS AFTER QUEUFS
 4088          024037                      BUFSIZ=AMODNM-START
 4089
```



```
 4090                                      ;ROUTINE TO DETERMINE WHETHER WRITE BUFFER ROTATION SHOULD TAKE PLACE,
 4091                                      ;AND TO DETERMINE CORE LIMITS OF BUFFER ROTATION, ALSO TO DETERMINE USE OF
 4092                                      ;RTI OR RTT INSTRUCTION.
 4093          027254                      .=IOQ
 4094  027252  012767  027270  150530  SETBUF: MOV     #1$,RESIV       ;SET UP A RESERVED INSTRUCTION TRAP.
 4095  027260  005046                      CLR     -(SP)               ;CLEAR A WORD ON THE STACK FOR A NEW PSW
 4096  027262  012746  027320              MOV     #2$,-(SP)           ;SET UP TO EXIT WITH RTT INSTRUCTION.
 4097  027266  000006                      RTT                         ;IF RTT NOT VALID IT WILL TRAP OUT.
 4098  027270  012767  000002  156410  1$: MOV     #RTI,TRCIA          ;TRAP COMES HERE, CHANGE RTT'S TO RTI'S.
 4099  027276  012767  000002  153632      MOV     #RTI,RTT1           ;  BY SETTING THE INSTRUCTION VALUE
 4100  027304  016767  152060  152066      MOV     CHKL2,CHKL1         ;CHANGE SET OF CHKSUM WORDS BECAUSE
 4101  027312  016767  152054  152062      MOV     CHKH2,CHKH1         ;OF RTI'S REPLACING RTT'S
 4102                                                                  ;OF ALL RTI'S TO THOSE OF RTI'S
 4103  027320  012767  177276  150462  2$: MOV     #RESINT,RESIV       ;RESTORE THE RESERVED INSTRUCTION TRAP VECTOR
 4104
 4105  027326  105067  172001              CLRB    MDL45               ;CLEAR THE PDP-11/45 INDICATOR
 4106  027332  012767  027352  150444      MOV     #3$,BUSEV           ;SET UP A NON-EXISTENT ADDRESS TRAP
 4107  027340  005737  177772              TST     @#PIRQRG            ;IS THE PROGRAMMED INTERRUPT REQUEST REGISTER THERE?
 4108  027344  112767  000001  151761      MOVB    #1,MDL45            ;IF IT IS, SET THE FLAG INDICATING AN 11/45
 4109  027352  105067  151725          3$: CLRB    KTPRES              ;CLEAR THE MEMORY MANAGEMENT INDICATOR
 4110  027356  012767  027402  150420      MOV     #4$,BUSEV           ;SET UP A NON-EXISTENT ADDRESS TRAP
 4111  027364  005037  177572              CLR     @#SSR0              ;IS THERE A KT-11 ON THIS SYSTEM?
 4112  027370  105167  151710              COMB    KTPRES              ;IF THERE IS, SET THE KT11 PRESENT INDICATOR
 4113  027374  116767  151704  151670      MOVB    KTPRES,KTSAV        ;SAVE FOR USE IN SECOND PASS OF CHAINING
 4114  027402  012767  015246  150374  4$: MOV     #BUSERR,BUSEV       ;REPLACE THE BUS ERROR TRAP VECTOR
 4115  027410  005067  150362              CLR     PSW                 ;RESET THE PSW
 4116  027414  105067  151673          STBFA: CLRB ROTI                ;ASSUME NO BUFFER ROTATION.
 4117  027420  105067  151671              CLRB    RELOC               ;ASSUME NO CODE RELOCATION
 4118  027424  016767  150350  151556      MOV     0,LOCORE            ;GET LOCORE ADDR. FROM LOC 0
 4119  027432  001004                      BNE     1$                  ;BP IF LINKED WITH CONFIG/LINKER PROGRAM
 4120                                                                  ;IF NOT ASSUMES NOT IN CHAIN MODE
 4121  027434  005167  151560              COM     LOCORE              ;INDICATE SETBUF HAS BEEN RUN
 4122  027440  000167  000420              JMP     SETXT               ;GO PREPARE TO EXIT SETBUF POUTINE
 4123  027444  004567  165152          1$: JSR     R5,TOP12S           ;GET TOP 12 BITS OF LOCORE PHYS ADDR
 4124  027450  001220                      LOCORE                      ;AND PUT IT INTO LCOR12
 4125  027452  032767  000777  151540      BIT     #777,LOCORE         ;IS IT ON A 256 WORD BOUNDARY ?
 4126  027460  001414                      BEQ     2$                  ;YES, CONTINUE
 4127  027462  042767  000777  151530      BIC     #777,LOCORE         ;NO, CLEAR OUT LOWER BITS
 4128  027470  042767  000007  151526      BIC     #7,LCOR12           ;ROUNDING OFF TO NEXT HIGHEST 256 BOUNDARY
 4129  027476  062767  001000  151514      ADD     #1000,LOCORE        ;ROUND UP TO THE NEXT 256 WORD BOUNDARY
 4130  027504  062767  000010  151512      ADD     #10,LCOR12          ;ALSO INDICATE THAT BOUNDARY IN LOCOR12
 4131  027512  016767  150324  151506  2$: MOV     42,HCOR12           ;ARE WE IN CHAIN MODE?
 4132  027520  001403                      BEQ     KTTEST              ;IF NO,GO DETERMINE IF KT IS TO BE USED
 4133  027522  112767  177777  151562      MOVB    #-1,CHN             ;IF YES, SET THE CHAIN INDICATOR
 4134  027530  105767  151550          KTTEST: TSTB KTPRES             ;IS THE KT11 AVAILABLE FOR USE?
 4135  027534  001035                      BNE     SETKT               ;IF YES, GO SIZE MEMORY USING IT
 4136  027536  012767  027600  150240      MOV     #5$,4               ;IF NO, ONLY SIZE MEMORY UP TO 28K
 4137  027544  012700  174000              MOV     #-4000,R0           ;INITIALIZE LOOP COUNTER TO POINT TO BANK 0
 4138  027550  005001                      CLR     R1                  ;INITIALIZE THE COUNTER OF 1K CHUNKS(BANKS)
 4139  027552  062700  004000          3$: ADD     #4000,R0            ;POINT TO THE FIRST WORD IN THE NEXT 1K BANK
 4140  027556  062710  000000              ADD     #0,(R0)             ;WILL TRAP IF LOC DOESN'T EXIST
 4141  027562  005201                      INC     R1                  ;IF NO, CONTINUE PROCESING AT 4$
 4142                                                                  ;IF YES,UP THE MEMORY BANK COUNT
 4143  027564  020127  000034              CMP     #28.,R1             ;HAVE 28 BANKS BEEN COUNTED?
 4144  027570  001370                      BNE     4$                  ;IF NO, GO CONTINUE SIZING
 4145  027572  062700  004000              ADD     #4000,R0            ;IF YES, INDICATE THE NUMBER OF WORDS IN R0
```

```
 4146  027576  000401                        BR      6$              ;SKIP OVER THE NEXT INSTRUCTION
 4147  027600  022626                5$:     CMP     (SP)+,(SP)+             ;IF THERE WAS NO LOCATION AT LAST TRY,
 4148                                                                  ;REMOVE THE TRAP FROM THE STACK
 4149  027602  000300                6$:     SWAB    R0              ;ISOLATE THE 12 HGH ORDER BITS, FIRST,
 4150                                                                  ;GET THE MOST SIGNIFICANT BITS IN THE LOW BYTE
 4151  027604  006300                        ASL     R0              ;ALIGN THE BITS TO ACCOUNT FOR
 4152  027606  006300                        ASL     R0              ;EXTENDED ADDRESS BITS (17 + 18)
 4153  027610  042700  170037               BIC     #170037,R0      ;ELIMINATE THE UNWANTED BITS FROM R0
 4154  027614  105767  151472               TSTB    CHN             ;ACT11 CHAIN?
 4155  027620  100054                        BPL     SETCOM          ;NO= DON'T ADD ROOM FOR MONITOR
 4156  027622  162700  000100               SUB     #100,R0         ;YES= ALLOW ROOM FOR MONITOR
 4157  027626  000451                        BR      SETCOM          ;GO TEST THE MEMORY
 4158  027630  004767  162160        SETKT:  JSR     PC,KTINIT       ;TURN ON THE KT11
 4159  027634  012737  077406  172304       MOV     #77406,@#KIPDR2
 4160  027642  012737  000001  177572       MOV     #1,@#SSR0       ;TURN ON KT11
 4161
 4162  027650  012737  027722  000004       MOV     #2$,@#4         ;SETUP TIMEOUT VECTOR
 4163  027656  016701  151342               MOV     LCOR12,R1       ;GET THE TOP OF THE PROGRAM
 4164  027662  062701  000040               ADD     #40,R1          ;ROUND IT UP
 4165  027666  042701  000037               BIC     #37,R1          ;MAKE SURE IT'S A 1K BOUNDARY
 4166  027672  010137  172344               MOV     R1,@#KIPAR2     ;LOAD THE KT11 REGISTER
 4167  027676  062737  000000  040000  1$:  ADD     #0,@#40000              ;TIME OUT IF THIS LOCATION IS NXM
 4168  027704  062737  000040  172344       ADD     #40,@#KIPAR2
 4169  027712  001371                        BNE     1$
 4170  027714  012700  007600               MOV     #7600,R0
 4171  027720  000403                        BR      3$
 4172  027722  022626                2$:     CMP     (SP)+,(SP)+
 4173  027724  016700  142414               MOV     KIPAR2,R0
 4174  027730  105767  151356        3$:     TSTB    CHN             ;CHAIN MODE?
 4175  027734  100006                        BPL     SETCOM          ;NO, JUST GO TEST MEMORY
 4176  027736  005067  151342               CLR     KTPRES          ;YES SHUT OFF KT FOR FIRST CHAIN PASS
 4177  027742  010067  151326               MOV     R0,HCR12A       ;SAVE THE REAL MEMORY SIZE
 4178  027746  000005                        RESET                   ;TURN KT OFF
 4179  027750  000667                        BR      KTTEST          ;NOW GO SIZE WITHOUT IT
 4180  027752  012767  015246  150024  SETCOM: MOV     #BUSERR,4       ;RESET BUS ERROR TRAP.
 4181  027760  010067  151242               MOV     R0,HCOR12       ;HI ADDR TO HCOR12.
 4182  027764  004767  000102        SETCM1: JSR     PC,TSTMEM       ;TEST MEMORY FOR BASIC FUNCTIONING
 4183                                                                  ;VIA ADDRESS UP/DOWN TEST
 4184  027770  004767  000626               JSR     PC,ANYPAR       ;CHECK FOR PARITY MEM. AND TURN IT ON
 4185  027774  016700  151226               MOV     HCOR12,R0       ;DIFF BETWEEN LCOR12 AND HCOR12
 4186  030000  010067  151224               MOV     R0,HCORSV       ;INITIALIZE SAVE LOCATION FOR KTON/OFF COMMANDS
 4187  030004  166700  151214               SUB     LCOR12,R0       ;MUST BE AT LEAST 40
 4188  030010  022700  000040               CMP     #40,R0          ;IS IT?
 4189  030014  101023                        BHI     SETXT           ;FORGET IT IF NOT.
 4190  030016  105267  151271               INCB    ROTI            ;ALLOW ROTATION.
 4191  030022  105767  151256               TSTB    KTPRES          ;NO RELOCATION IF NO KT11
 4192  030026  001413                        BEQ     1$
 4193  030030  022700  000400               CMP     #400,R0         ;IF 8K OR GREATER, ALLOW CODE RELOCATION
 4194  030034  101010                        BHI     1$              ;BRANCH IF NOT
 4195  030036  016767  151164  151174       MOV     HCOR12,OFFMX    ;CALCULATE THE HIGHEST LOC TO WHICH
 4196  030044  166767  151154  151166       SUB     LCOR12,OFFMX    ;THE PROGRAM CAN BE RELOCATED
 4197  030052  105267  151237               INCB    RELOC           ;ENABLE RELOCATION
 4198  030056  004767  174610          1$:  JSR     PC,LOADWC       ;LOAD WORST CASE UNIBUS NOISE PATTERN
 4199  030062  000005                        RESET                   ;TURN OFF KT11 IF IN USE
 4200  030064                        SETXT:
 4201  030064  005067  147710               CLR     0               ;CLEAR LOC 0.
```



```
 4202  030070  000205                        RTS     R5              ;EXIT.
```

```
                                ;MEMORY ADDRESS UP/DOWN TEST
                        
030072  026767  151130  151124  TSTMEM: CMP     HCOR12,LCOR12   ;ANY MEMORY TO TEST ?
030100  001001                  BNE     5$              ;YES, CONTINUE
030102  000207                  RTS     PC              ;NO, RETURN
030104  016702  151114          5$:     MOV     LCOR12,R2       ;LOAD 12 BIT LOCORE ADDRESS
030110  016701  151104          MOV     LOCORE,R1       ;R1 CONTAINS LOCORE PHYS ADDRESS
030114  012767  030476  147662  MOV     #NOMEM,4        ;LOAD TIME-OUT VECTOR
030122  105767  151156          1$:     TSTB    KTPRES          ;USE KT11?
030126  001002                  BNE     2$              ;YES- BRANCH
030130  010100                  MOV     R1,R0           ;NO- SET VA EQUAL TO PA
030132  000404                  BR      3$
030134  012700  040000          2$:     MOV     #40000,R0       ;SET VA TO PAGE 2
030140  010267  142200          MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL ADDRESS
030144  012703  000040          3$:     MOV     #32.,R3         ;COUNT 1 BLOCK
030150  010120                  4$:     MOV     R1,(R0)+        ;LOAD LOCATION WITH ITS ADDRESS
030152  062701  000002          ADD     #2,R1
030156  005303                  DEC     R3
030160  001373                  BNE     4$
030162  005202                  INC     R2              ;AFTER 1 BLOCK, UP PAGE ADDRESS FIELD
030164  020267  151036          CMP     R2,HCOR12       ;AT TOP?
030170  002754                  BLT     1$              ;NO- CONTINUE
030172  005302                  DEC     R2              ;YES- TEST GOING DOWN
030174  162701  000002          SUB     #2,R1
030200  105767  151100          TSTM1:  TSTB    KTPRES          ;USE KT11?
030204  001002                  BNE     1$              ;YES- BRANCH
030206  010100                  MOV     R1,R0           ;NO- LOAD VIRTUAL ADDRESS POINTER WITH PHYS ADDRESS
030210  000404                  BR      2$
030212  012700  040076          1$:     MOV     #40076,R0       ;MAP VA TO TOP OF 1ST BLOCK IN PAGE 2
030216  010267  142122          MOV     R2,KIPAR2       ;MAP PAGE 2 TO PA
030222  012703  000040          2$:     MOV     #32.,R3         ;COUNT 1 BLOCK
030226  011067  150764          3$:     MOV     @R0,MEMWAS
030232  020167  150760          CMP     R1,MEMWAS       ;DOES LOCATION CONTAIN ITS PA?
030236  001120                  BNE     MERR            ;NO- BRANCH
030240  162700  000002          4$:     SUB     #2,R0
030244  162701  000002          SUB     #2,R1
030250  005303                  DEC     R3
030252  001365                  BNE     3$
030254  005302                  DEC     R2              ;AFTER CHECKING 1 BLOCK, ADDRESS NEXT
030256  020267  150742          CMP     R2,LCOR12       ;DONE?
030262  002346                  BGE     TSTM1           ;NO- CONTINUE CHECKING
030264  016702  150736          MOV     HCOR12,R2       ;YES- SET TO LOAD MEMORY
030270  010201                  MOV     R2,R1           ;FROM TOP DOWN
030272  005302                  DEC     R2
030274  006301                  ASL     R1
030276  006301                  ASL     R1
030300  006301                  ASL     R1
030302  006301                  ASL     R1
030304  006301                  ASL     R1
030306  005301                  DEC     R1
030310  006301                  ASL     R1
030312  105767  150766          TSTM2:  TSTB    KTPRES          ;USE KT11?
030316  001002                  BNE     1$              ;YES- BRANCH
030320  010100                  MOV     R1,R0           ;NO- LOAD VIRTUAL POINTER WITH
                                ;PHYSICAL ADDRESS
030322  000404                  BR      2$
```



```
030324  012700  040076          1$:     MOV     #40076,R0       ;LOAD VIRTUAL WITH TOP ADDRESS OF
                                ;1ST BLOCK IN PAGE 2
030330  010267  142010          MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL
030334  012703  000040          2$:     MOV     #32.,R3         ;COUNT A BLOCK
030340  010110                  3$:     MOV     R1,@R0          ;LOAD LOCATION WITH ITS PHYS ADDRESS
030342  005110                  COM     @R0             ;COMPLEMENT IT
030344  162701  000002          SUB     #2,R1
030350  162700  000002          SUB     #2,R0
030354  005303                  DEC     R3
030356  001370                  BNE     3$
030360  005302                  DEC     R2              ;AFTER LOADING A BLOCK, SET UP FOR NEXT
030362  020267  150636          CMP     R2,LCOR12       ;DONE?
030366  002351                  BGE     TSTM2           ;NO- BRANCH
030370  005202                  INC     R2              ;YES- SETUP TO CHECK IT
030372  062701  000002          ADD     #2,R1
030376  010104                  MOV     R1,R4
030400  105767  150700          TSTM3:  TSTB    KTPRES          ;USE KT11?
030404  001002                  BNE     1$              ;YES- BRANCH
030406  010400                  MOV     R4,R0           ;SET VIRTUAL ADDRESS EQUAL TO PHYS ADDRESS
030410  000404                  BR      2$
030412  012700  040000          1$:     MOV     #40000,R0       ;SET VIRTUAL ADDRESS TO START OF PAGE 2
030416  010267  141722          MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL ADDRESS
030422  012703  000040          2$:     MOV     #32.,R3         ;COUNT A BLOCK
030426  011067  150564          3$:     MOV     @R0,MEMWAS
030432  010401                  MOV     R4,R1
030434  005101                  COM     R1
030436  020167  150554          CMP     R1,MEMWAS       ;DOES LOCATION CONTAIN COMPLEMENT OF ITS PHYS ADDRESS?
030442  001016                  BNE     MERP            ;NO- BRANCH
030444  005720                  4$:     TST     (R0)+
030446  062704  000002          ADD     #2,R4
030452  005303                  DEC     R3
030454  001364                  BNE     3$
030456  005202                  INC     R2              ;CHANGE 12 BIT ADDRESS AFTER CHECKING A BLOCK
030460  020267  150542          CMP     R2,HCOR12       ;DONE?
030464  002745                  BLT     TSTM3           ;NO- BRANCH
030466  012767  015246  147310  MOV     #BUSERR,4       ;RESET THE TIME-OUT TRAP VECTOR
030474  000207                  RTS     PC              ;YES- RETURN

030476  022626                  NOMEM:  CMP     (SP)+,(SP)+             ;POP THE STACK POINTER TWICE
030500  012767  015246  147276  MERR:   MOV     #BUSERR,4       ;RESET THE TIME-OUT TRAP VECTOR
030506  105267  150623          INCB    MEMERF
030512  010067  150466          MOV     R0,MEMVA        ;SAVE FAILING ADDRESS
030516  104413  001204          GETPAS,MEMVA
030522  010167  150466          MOV     R1,MEMSBE       ;SAVE EXPECTED CONTENTS
030526  016746  150454          MOV     MEMPA,-(SP)     ;CONVERT PHYSICAL ADDRESS TO ASCII
030532  016746  150452          MOV     MEMEA,-(SP)
030536  004567  163410          JSR     R5,OACNVX
030542  026365                  MERR2
030544  016746  150444          MOV     MEMSBE,-(SP)    ;CONVERT SHOULD BE TO ASCII
030550  004567  163364          JSR     R5,OACNV
030554  026400                  MERR3
030556  016746  150434          MOV     MEMWAS,-(SP)    ;CONVERT WAS VALUE TO ASCII
030562  004567  163352          JSR     R5,OACNV
030566  026413                  MERR4
030570  004567  164026          JSR     R5,TOP12S       ;GET NEW HIGH CORE VALUE
030574  001206                  MEMPA
```

```
 4315  030576  016767  150410  150422          MOV     MEM12,HCOR12
 4316  030604  042767  000007  150414          BIC     #7,HCOR12
 4317  030612  162767  000040  150406          SUB     #40,HCOR12
 4318  030620  000207                          RTS     PC
```



```
 4319                                  ;THIS ROUTINE CHECKS FOR PARITY MEMORY
 4320
 4321  030622  105067  150512          ANYPAR: CLRB    PPRES           ;CLEAR PARITY PRESENT FLAG
 4322  030626  105067  150507                  CLRB    CNT1            ;CLEAR # OF REGISTERS COUNTER
 4323  030632  012767  030654  147150          MOV     #1$,RESIV       ;SET UP FOR A RESERVED INSTR, TRAP
 4324  030640  076600  000101                  MED,RDSERV              ;IS THIS AN 11/60 ?
 4325  030644  152767  000002  150466          BISB    #BIT1,PPRES     ;YES, SET 11/60 PARITY PRESENT FLAG
 4326  030652  000401                          BR      11$             ;GET OUT
 4327
 4328  030654  022626                  1$:     CMP     (SP)+,(SP)+             ;POP STACK POINTER TWICE
 4329  030656  012767  015276  147124  11$:    MOV     #RESINT,RESIV   ;RESTORE RESERVED INSTR, TRAP VECTOR
 4330  030664  012701  172100                  MOV     #172100,R1      ;LOAD ADDRESS OF 1ST REGISTER
 4331  030670  012767  030714  147106          MOV     #3$,BUSEV       ;LOAD TIME-OUT VECTOR
 4332  030676  012702  001430                  MOV     #PAPTAB,R2      ;LOAD ADDRESS OF TABLE OF ADR'S
 4333  030702  005011                  2$:     CLR     (R1)            ;IF REGISTER IS THERE - CLEAR IT
 4334  030704  105267  150431                  INCB    CNT1            ;IT'S HERE, COUNT IT
 4335  030710  010122                          MOV     R1,(R2)+        ;PUT IT'S ADDRESS INTO THE TABLE
 4336  030712  000401                          BR      4$              ;CONTINUE
 4337
 4338  030714  022626                  3$:     CMP     (SP)+,(SP)+             ;POP THE STACK POINTER TWICE
 4339
 4340  030716  062701  000002          4$:     ADD     #2,R1           ;STEP TO THE NEXT ADDRESS
 4341  030722  022701  172136                  CMP     #172136,R1      ;ALL REGISTERS CHECKED ?
 4342  030726  002365                          BGE     2$              ;NO, DO THE NEXT ONE
 4343  030730  105767  150405                  TSTB    CNT1            ;YES, FIND ANY PARITY REGISTERS ?
 4344  030734  001403                          BEQ     5$              ;NO, GET OUT
 4345  030736  152767  000001  150374          BISB    #BIT0,PPRES     ;YES, SET PARITY PRESENT
 4346
 4347  030744  012767  015246  147032  5$:     MOV     #BUSERR,BUSEV   ;RESET TIME-OUT VECTOR
 4348  030752  012767  015672  147134          MOV     #PARERR,PARVCT  ;RESET PARITY VECTOR
 4349  030760  000207                          RTS     PC              ;RETURN
 4350
 4351          000001                          .END
```

```
AAT     022747          1405    3583#
ACRLF = 022772           955     970     973    1360    1374    2718    3548    3561    3587#
ACSR  = 000052           167#
ACSRAC  026136           921    3951#
ACSRC   026152           915    3953#
ADDRHI  001044           267#   1726    2171*
ADDRLO  001042           266#   1725    2116    2170*
ADRERR  013510          2168    2175#   2193    2195
ADTERR  026177           946    3958#
ADTEX   026311           953    3975#
ADTE2   026233           853     889    3964#
ADTE3   026220           851     891    3962#
ADTE4   026265           842     897    3970#
ADTE4A  026275           945    3972#
ADTE5   026250           849     903    3967#
ADTE6   026205           831     909    3960#
ADTE7   026317           859    3977#
AEND    025721           765     931     938    1000    1028    1035    1096    1103    3919#
AEND1   025733           998    3921#
AEND2   025746           996    3923#
AERCT   026436           798    3994#
AERNM   026423           795     804    3992#
AERNMB  026335           825     885    3981#
AERROR  026130           949    3950#
AERRS   025657          1964    3910#
AERTOT  026470           801    3999#
ALCLR   001364           402#   1716
ALLBK   001354 G          36#    396#    579    1044    1054
ANDSTA  025621          1958    3903#
ANODNM  025573           344    1952    1968    2226    2230    2275    3898#   4088
ANYPAR  030622          4184    4321#
APASS   025766          1004    3927#
APC     025605          1955    3900#
APCLR   001366           403#   1720
APSCNT  025640          1961    3907#
APSEND  026007          1097    1104    3931#
APTGAL= ****** U           1      27      28      29     206     318     435     460     462     535     560    1773    1779
                        2829    3573    3580    4201
ASB   = 000056           171#   1197    1600*   1632*
ASTAT = 000054           169#
ASTATC  026167           913    3956#
ATE     001342           386#   2292*   2295*   2298*   2307    2320    2330    2341*
AWAS  = 000060           172#   1195    1212    1601*   1631*
BDCNV   014372 G          32#    797     800     824     858     884    1003    1090    1960    1963    2352#   2878    2895
                        2919    2976    2981    2985
BIT0  = 000001            31#   1432    1714    2054    2065    2330    2341    2668    3021    3046    3097    3104    3114
                        3140    3162    3203    3413    4345
BIT1  = 000002           130#   1718    1756    1918    2069    2619    2652    2674    2997    3027    3048    3106    3164
                        3205    4325
BIT10 = 002000           121#    598     608    1595    1792
BIT11 = 004000           120#    598     603    1788
BIT12 = 010000           119#    598     618    1046    1086    1798    3071    3142    3269
BIT13 = 020000           118#    694     758     802     922    1020    1121    1154    1781    1784    3223    3268
BIT14 = 040000           117#    697     979    1119    1782    1947    1989    1994    3221
BIT15 = 100000           116#
BIT2  = 000004           129#
```



```
BIT3  = 000010           128#
BIT4  = 000020           127#
BIT5  = 000040           126#
BIT6  = 000100           125#    143
BIT7  = 000200           124#   2298    3099
BIT8  = 000400           123#
BIT9  = 001000           122#   1152    3538
BKCNT   001025           252#   1052    1155*   1797*
BKPTR   001270           350#    470*    616*    665    1753*   3116    3286    3467*
BKQSVC  003140           570     716#
BKQUE   001012           241#
BKTIME  001336           381#   1165
BKTIMR  001034           261#   1162*   1165*
BKTYM   020042           617    3097#
BPSCNT  026016          1091    3933#
BRAKE   001013           242#    577     615*   1041*   1059    1151*   1438*   2610*   2639*   3288*   3516*   3554*
BREAK$= 104407           684    2549#
BREAK,  003454           686     784#
BUFSIZ= 024037          4088#
BUSERR  015246           188     499    1887    2025    2029    2579*   3037    4114    4190    4294    4298    4347
BUSEV   000004           188#    462     464*    469*   4106*   4110*   4114*   4331*   4347*
CADDR   001512           441     443#    756*    777*    790*    812*    873*   1079*   1692*
CAST  = ****** G          35#    210
CCSRA   001160           307#    488     905*    917*    981*    984     987     988*
CDATA$= 104414          2559#
CDATA,  007636          1529#   2558
CDATA1  007662          1535#   1652
CDERR   001516           447#   1606
CDERR1  001520           448#   1535*   1536*   1605
CDFLG   001156           306#   1531*   1555    1651*
CDRET   010464           449    1647#
CHKH1   001402           416#   4101*
CHKH2   001372           412#   4101
CHKL1   001400           415#   4100*
CHKL2   001370           411#   4100
CHKSUM  001374 G          34#    413#
CHN     001312           363#    535     590    1109    1397    1777    1860    2725    2734    2742*   2755    4133*   4154
                        4174
CHNOUT  016104          1411    2663    2725#
CKCONT  007642          1528    1531#
CKHUNG  020462          1084    3193    3202#
CLKADR  001420           423#   3171*   3184*   3408    3415*
CLKHLD  001422           424#   3172*   3184    3407    3414*
CLKOFF  020334          1066    2600    2627    2667    3160#
CLKON   020416          1068    2754    3182#
CLOCK   001353 G          33#    395#    924    1021    1082    1092    1402    1714    1718    3018*   3021*   3027*   3036*
                        3046    3048    3104    3106    3160    3162    3164    3182    3203    3205    3275*
CLOCKL  001356 G          32#    398#   3019    3022    3059    3110    3167    3209
CLOCKP  001360 G          33#    399#   3025    3028    3063    3112    3169    3207
CLRQUS  014765           501     529    1421    1427    1755    2483#   3421
CLRVEC  ------           530    1754    2500#
CLR40   ------           526    3651#
CNT1    ------           178#   1932    2078    2686    4322*   4334*   4343
COFF    ------          1498#   3859
COFFF   023652           3101   3669#
COMCON  010646          1663    1691#   1742    1746    1842    1942    1990    2028    2135    2727
```

```
COMCO1  010642            1673    1690#
COMCO2  010646            1692#   3862
COMCO3  010656            1395    1695#   2001    2010    2031    2036    2044    2048    2082    2094    2102    3307
COMCO4  010662            1696#   1702
COMQUE  001502             441#    781     989    1157    1686
COMTAB  025106            1666    3810#
CON     013130            2086#   3856
CONN    023643            2093    3666#
CONTRL  001476             431#   1766    1768*   1920*   2071*   2092*   2100*
CONVRT  014176            2293    2302#
CPUERR  001470             428#
CR      022742            1098    1407    3581#
CRCR    022744             764     928     937    1025    1034    1095    1102    3278    3279    3281    3294    3295    3298
                          3582#
CSRA  = 000050             166#    987*   1230
CSTART  001514             444#    755*    757     778*    793*    815*    869*    874     986    1080*   1416*   1420*   1422*
                          1683*   3119    3192*   3217
CTRLC   022770            1440    3588#
CTRLCA  007044            1318    1387#   2651
CTRLCB  007050            1073    1388#   3259
CTRLCD  007200            1398    1411#
CTRLX   007204            1386    1413#   2601    2637
CXRET   001262             347#
DATCK$= 104417            2565#
DATCK.  007626            1527#   2564
DATER$= 104405            2545#
DATFLG  001347             391#   1527*   1529*   1639
DDPPTR  000042             205#
DECODE  010512            1375    1656#
DECVAL  026513            2354    4003#
DERRCT  001130             295#   1532#   1568*   1592*   1593    1603    1623    1631    1641
DERRX$= 104421             447    2569#
DERRX.  003562             743     810#
DERRX1  003630             820#    822
DES     012500            1979#   3825
DIRA    012334            1408    1941    1944#
DIRIND  001011             240#   1939*   1949    1965
DOT     022737             534    1695    1835    2623    3580#
DRPHNG  020754            3240    3268#
DSTADR  001054             271#    625*    627*    689*    692     712     716*    785*   1625*   1638*
EMTV    000030             199#
ENDCOM  004670             762     818     875     990#   1019    1088
ENDPS$= 104402            2539#
ENDSVA  005022             982    1020#
ENDSVB  005214            1057    1061#   1806
ENDSVC  005014             742    1018#
ENDSVD  005304            1053    1058    1060    1072    1074#   1153
ENDSVE  005104            1039#   1130    1133    1148    1156
ENDSVF  005272            1051    1071#
ENDSVG  005160            1045    1052#   3195
END$  = 104403            2541#
ERCNT = 000030             158#    823     876*    878*    883     977    1627*   1630*   1780*   1962
ERRIND  001017             246#    810*    864*    866*    868*    870     886     942     951     957
ERRLIN  001254             343#    977
ERRNMB  026327             933     940    3979#
ERRN$ = 104410            2551#
```



```
ERROR$= 104404            2543#
ERRPOS  001154             305#    833*    834*    837     845     854
ERRSVC  004042             742     868#
ERRTYM  020052             926    1023    3099#   3291
ERSVCA  004050             865     867     869#
ERSVCB  004116             877     879#
ERSVCC  004122             880#    882
ERSVC1  004034             742     866#
ERSVC2  004024             743     864#
ERSVND  004576             806     923     958     975#
ERSVX   004332             860     910     922#
EXIT$ = 104400            2535#
EXIT.   005736            1173#   1241    2534
EXITT.  005740            1170    1174#   1248    1309
EXIT2.  005774            1185#   1201    1223    1251    1305
FAKESR  001004             234#    467     506
FATAL   024100            3567    3695#
FILCNT  001306 G            35#    359#   1512    2132
FILCTR  001020             247#   1512*   1514*   1518*
FILL    013214            2109#   3831
FILLER  001307 G            35#    360#   1515
FILLID  001324             370#   1694*   2109*   2133
FILLNM  013634             794     999    1951    2211#   3289
GETADR  013450            1723    2114    2163#
GETNAM  013662            1985    2111    2222#
GETNUM  013516            2123    2163    2181#
GETPA$= 104413             893     899     990    2557#   3557    4301
GETPA.  021272            2556    3343#
GTOK    010554            1666#
GTOKX   010560            1667#
GTOK1   010562            1668#   1676    1678
GTOK2   010604            1669    1677#
GWBUF$= 104412            2555#
GWBUF.  021150            2554    3312#
HARDSR= 177570              76#    463
HCORSV  001230             332#   2018    4186#
HCOR12  001226 G            32#    330#   1854    1859    2006    2008*   2018*   2738*   2793    2858    2871    3772    3785
                          4131*   4181*   4185    4195    4205    4223    4244    4292    4315*   4316*   4317*
HCR12A  001274             352#   2738    2739    4177*
HLT     023746            1262    1294    2528    2583    3568    3678#
HMS     017310 G            33#   2947#   3076    3146
HNGFLG  001344             388#
HNGTIM  001362 G            34#    401#   3237
HOLD5   025667            2879    2880    2886    2896    2898    2908    2920    2922    2931    2977    2978    2979    2982
                          2983    2986    2987    3913#
HOLD6   025675             965     966    2142    2143    2157    2158    2713    2714    3915#
HOLD8   025706            2150    2151    3917#
HOP     014166            2296    2300#
HOURS   001405 G            33#    418#   2978*   2979*   3079    3080    3081
HUNGCK  020434 G            34#   3190#
IE    = 000100             143#    543    1347    1400
INIT  = 000022             155#    625
INPUT   002162             536     540#   1696    1837
INVADR  023344            2175    3632#
INVCD1  023131            1662    3606#
INVCD2  023203            1741    3614#
```

```
INVCD3  023241          1745    3620#
INVCD4  023316          1841    3628#
INVCMD  023107          1690    3602#
INVNAM  023370          2264    3636#
INVTRP  023040          2527    3595#
IOBKID  001060           273#    558     583#    592#    599     610     633     635#    637     639#    653     660#    1811#
                        2609#   2638#   3515#   3553#
IOQ     027252           148     678     1292    1298    2488    4017#   4093
IOQL  = 000620           145#    148     2493    4017
IOQLIM= 030072           148#    676     1296
IOQSVA  003040           694#    722
IOQSVB  003122           710#
IOQSVC  002742           557     676#
IOQSVD  003106           704#   1645
IOQSVE  003032           632     691     693#    786
IOQSVN  003016           442     689#
IOQUE   001006           237#    555     679#    1166    1288    1300#   2483    2484
IOQ1    001074           279#   1290    1292#   1296    1298#   1299#   1301#   2489#
IOQ2    001076           280#    676     678#    680     681#   1290    2490#
IOTV    000020           194#
ITOA    013426          2119    2121    2155#   2612    2856
ITOAX   013336          2138#   2866    3547    3560
ITOA8   013372          2146#
ITOUT   013444          2144    2152    2159#
KAY     024503          2887    3742#
KBOFLO  023505          1379    3649#
KBPTR   001036           264#    537#    542#   1324    1343#   1344#   1354    1356#   1357    1361#   1366    1372#   1373#
                        1665    1684#   2181    2206#   2222    2270#   2282#
KBSRVC  006552           213    1314#
KBUF    026712           542    1324    1354    1361    1366    1373    4009#
KBUFL = 000040           144#   1324    4010
KIPAR0= 172340           100#   1581    3358    3533#
KIPAR1= 172342           101#    487#
KIPAR2= 172344           102#   3438#   3442#   3443    3445#   3446#   3447    3448#   3453#   3458    3493#   3498#   3506#
                        3779#   4166#   4168#   4173    4216#   4232#   4261#   4280#
KIPAR3= 172346           103#   3439#   3447#   3449#   3454#   3455#   3456#   3462    3475    3494#   3495    3499#   3502
KIPAR7= 172356           104#   1912#
KIPDR0= 172300            95#   1903
KIPDR1= 172302            96#   3524
KIPDR2= 172304            97#    480#   3423#   4159#
KIPDR3= 172306            98#   3424#
KIPDR7= 172316            99#   1913#
KLOCK   017550           511    3018#
KONTRL  001500           432#   1766#   1767#   1768    2091#   2092    2099#   2100
KTDIS   024007          2009    3685#
KTERR   015254          1874    2581#
KTFATL  024050          2582    3691#
KTINIT  012014           479     514    1430    1874#   2019    2748    4158
KTOFF   012562          1999#   3834
KTON    012632          2014#   3837
KTONM   024615           516    2023    3757#
KTPRES  001304 G          36#    356#    477     512     531    1428    1575    1610    1743    1999    2002#   2017#   2605
                        2628    2642    2728    2743#   2744    3347    3404    3776    4109#   4112#   4113    4134    4176#
                        4191    4211    4227    4254    4275
KTSAV   001272           351#   2736    2743    4113#
KTSET0  012150          1901#   3412
```



```
KTTEST  027530          4132    4134#   4179
KW11L = ****** G          33#    398
KW11P = ****** G          33#    399
KYBDV   000060           213#
LCLEAR= ****** G          34#    402
LCOR12  001224 G          32#    329#   1852    3442    3501    3770    4128#   4130#   4163    4187    4196    4205    4209
                        4242    4270
LDBRK,  006366          1274#   2548
LDIOQ   006402          1275    1283#
LDTYPQ  006256          1191    1194    1196    1198    1205    1208    1210    1217    1227    1236    1244    1249    1253#
LGHOLD  001474           430#   1758#   1763    2654#   2658
LOADWC  024672           533    3513    3769#   4198
LOCK    001327           373#   1143    1707#   1710#   2765    2802
LOCORE  001220           327#    471    1849    2855    3774    4118#   4121#   4124    4125    4127#   4129#   4209
LOGBUF  031422          2709    2999    4085#
LOGIC   016134           207    2732#
LOGICA  016144          2662    2734#
LPB     001174           317#   3304
LPON    021110          3303#   3853
LPS     001172           316#   3303
LPWK1   021144          1391    3308#
LPWK2   021146          1392    3309#
MAP     012314          1939#   3828
MAPREG= 170200            81#
MAPRET  001252           342#   3418#   3521
MAP22S= 104415          2561#
MAP22,  021426          2560    3384#
MAXTIM  001426           426#   3214#   3232    3235#   3237
MDIRE   025627          1940#   1967#   3905#
MDL45   001333           378#   1891    4105#   4108#
MDXCNT  001022           249#    620#    645#    659    1048#   1137    2787    3271#
MDXCTR  001021           248#    503#    640#    659#   2777    2783    2787#   3323    3328#
MED   = 076600            82#   1762    1764    2088    2657    2659    3003    4324
MEMCHG  023724          2850    3675#
MEMEA   001210           323#   3544    3546    3559    4304
MEMERF  001335           380#   2847    2851#   4299#
MEMERR  001472           429#
MEMPA   001206           322#   3543    3545    3558    4303    4314
MEMSBE  001214           325#   4302#   4307
MEMVA   001204           321#   3555#   3557    4300#   4301
MEMWAS  001216           326#   4234#   4235    4282#   4285    4310
MEM12   001212           324#   4315
MERR    030500          4236    4286    4298#
MERR1   026344          2849    3983#
MERR2   026365          3986#   4306
MERR3   026400          3988#   4309
MERR4   026413          3990#   4312
MFLAG   001343           387#
MINITS  001410           419#   2983#   3082    3083    3149    3150
MMINS   026121          3149#   3150#   3948#
MOD     013222          2110#   3819
MODADR  001136           298#    700#   1175    1192    1206    1220    1637
MODCNT  001015 G          36#    244#   1049#   1052    1056    1071    1794#   1802    1808    1825    1829    3257    3272#
MODCTR  001016 G          36#    245#    588     631#   1808#
MODEND  025775          1029    1036    3929#
MODPTR  001040           265#    594#    595     616     665#    668#    718    1810#   1988    2117    2224#   2281#
```

```
MODQ    026522 G      32#   470   668  1115  1753  1770  1810  1944  1991  2272  3120  3213  3243
                    4008#
MODREA  001124        293#  1550* 1562#
MODRPA  001122        292#  1569* 1574
MODR12  001126        294#  1583
MODWEA  001116        290#  1553* 1565* 1598* 1600
MODWPA  001114        289#  1570* 1572  1597* 1601
MODW12  001120        291#  1582
MORE    010576       1674#
MOVBNK  001250        341#  3453  3454  3458*
MOVFLG  001345        389#  3419* 3433* 3532
MOVING  001325        371#  2603  3425  3430* 3459* 3504* 3510*
MSCONT  003326        751    754#
MSECS   026124       3151* 3152* 3949#
MSGDS = 104422       1635  2571#
MSGD.   003466        743    790#
MSGD1.  003556        805    806#
MSGFLG  001346        749    750*  752*  760
MSGNS = 104411       2553#
MSGN.   003322        743    752#
MSGSS = 104416       2563#
MSGS.   003312        743    750#
MSGS  = 104406        505    508   516   520   526   534  1261  1262  1293  1294  1346  1360  1374
                     1379   1401  1405  1406  1407  1440  1662  1690  1695  1741  1745  1835  1841
                     1968   2009  2023  2027  2043  2057  2059  2067  2081  2093  2101  2143  2151
                     2158   2175  2264  2527  2528  2547* 2582  2583  2611  2621  2623  2640  2650
                     2676   2704  2849  2850  2854  2885  2886  2887  2907  2908  2930  2931  3034
                     3540   3556  3567  3568
MSG.    003426        743    776#
MSG1.   003444        759    771   780#
MVCODE  021474        528   1067  1390  1751  1839  2607  2753  3404#
MVCODS  021564       3420#
MVCOD1  021574       3422*  3519
MVCOD2  022040       3441   3451  3457# 3496  3503
MVCOD3  022312       3497   3504#
MVDOWN  001330        374#  3440* 3450* 3511
MVERR   022564       3479   3487  3553#
MVMAP   022412       3434   3518  3521#
MVMSG   022366       3512   3515#
MVRTRY  022060       3460#  3566
MXWBF   001256        344#  1859* 1862* 1863* 1864* 1865* 1866* 1867* 1868* 1870* 2832* 2833* 2834*
                     2835* 2836* 2837* 2838* 2840* 3320  3322
NEW   = ****** U       28    37   152   321   343   823   870   876   883   911   957   977  1160
                     1627  1780  1962  2537  2541  2549  3573  3766  3907  3956  3979
NOCLK   024317       3034  3721#
NOKT    023776        520  2027  3683#
NOMEM   030476       4210  4297#
NOPAR   024021       2059  2067  3687#
NOSWR   024633        508  3760#
NSTOP   001352        394#   651* 1145  1812* 2769  2804
NUMBEA  001052        270#   895   901   992
NUMBER  001046        268#   735* 817*  892*  893   898*  899   990   993
NUMBPA  001050        269#   894   900   991
NUMPWR  025551       2930  3894#
NUMSYS  025531       2907  3891#
OACNV   014140 G      32#   850   852   888   890   912   914   964   995  1954  1957  2156  2291#
```



```
                     2702  2712  4308  4311
OACNVX  014152        830   841   848   896   902   908   920   997  2141  2295#  2699  4305
OACNVR  014160       2149  2298#
OASCEA  001112        286#  2291* 2300* 2322  2332  2337
OFFMX   001240        336#  1063  1739  2739* 4195* 4196*
OFFSET  001242 G       34#   337#  601   605  1061  1786  1790  1804  1846  1853  2752  2790  2796
                     2818  2832  3429* 3431  3436  3438  3448  3455  3491  3505* 3506  3507  3509
                     3522  3533  3542  3771
OFFSTD  001244        338#   527* 1061* 1062* 1063  1065* 1389* 1711* 1747* 1838* 2752* 3409  3416*
                     3431  3435
OFFSTM  001246        339#  3429  3435* 3436  3439  3449  3456  3491  3495  3500  3505
OPEN  = 000000        115#   204   208   209   234   237   238   239   240   241   242   243   244
                      245   246   247   248   249   250   251   252   253   254   255   256   258
                      260   261   262   264   265   266   267   268   269   270   271   272   273
                      274   276   276   277   278   279   280   281   282   283   284   285   286
                      289   290   291   292   293   294   295   296   297   298   299   300   301
                      302   303   304   305   306   307   321   322   323   324   325   326   327
                      329   330   332   334   335   336   337   338   339   341   342   346   347
                      348   349   350   356   361   362   363   364   365   366   367   370   371
                      372   373   374   375   378   379   380   384   385   386   387   388   389
                      390   391   393   394   395   396   419   420   421   422   423   424   425
                      426   428   429   430   443   444   448  4020  4021  4022  4023  4024  4025
                     4026  4027  4028  4029  4030  4031  4032  4033  4034  4035  4036  4037  4038
                     4039  4040  4041  4042  4043  4044  4045  4046  4047  4048  4049  4050  4051
                     4052  4053  4054  4055  4056  4057  4058  4059  4060  4061  4062  4063  4064
                     4065  4066  4067  4068  4069  4070  4071  4072  4073  4074  4075  4076  4077
                     4078  4079  4080  4081  4082  4083
PAENBL  012232       1434  1918# 2056
PAMSG   015776       1928  2692  2697#
PAMSG1  016032       2622  2677  2708#
PAOFF   024034       2081  3689#
PARERR  015672        222  2510  2666# 4348
PARTAB  001430        427#  1922  2073  2680  4332
PARVCT  000114        222#  2510* 2511* 4348*
PASEND  005310        741  1078#
PASREL  001033 G       36#   260#  1139  1825* 1826* 1827* 1828* 1831*
PASTOT  001032        258#  1134* 1139
PASTYM  020062       1085  3101#
PCLEAR= ****** G       34#   403
PCONT   025357       2703  3871#
PDP70 = ****** U        1    27    28    29   288   369   427   501   518   522   528   829   893
                      990   997  1066  1390  1432  1571  1582  1727  1751  1756  1839  1845  1912
                     1916  2004  2022  2052  2086  2104  2465  2594  2599  2607  2618  2619  2649
                     2652  2664  2751  2753  2779  2790  2796  2798  2818  2832  2871  2994  3351
                     3381  3409  3414  3541  3557  3560  3573  3580  3640  3666  3672  3770  3785
                     3860  3921  3951  3960  3967  3986  4110  4162  4184  4200  4301  4305  4319
PDREND  001302        355#  1885
PDRTAB  001276        353#  1876
PENABL  023756       2057  3680#
PERR1   025330       2704  3866#
PERR2   025427       2676  3879#
PIRQRG= 177772         90#  4107
PIRQS = 000004        114#   454  1320
PIRQ.   006416        394  1287#
POFF    013044       2065# 3843
PON     013004       2047* 3840
```

```
PONN    013014            522    2047    2054#
POPSP = 005726            140#   1199    1211    1239    1245    1250    1319    1349    1359
POPSP2= 022626            141#
PPRES   001340            384#   1432    1756    1918    2054    2065    2069    2619    2652    2668    2674    2997    4321#
                         4325#   4345#
PREG    025350           2700    3869#
PRTY4 = 000200            136#    216    3306
PRTY5 = 000240            135#
PRTY6 = 000300            134#
PRTY7 = 000340            133#    189     191     195     198     214     223    1283    1413    1875    1888    2511
PS    = 177776             74#   1889*   1893*   1895*
PSCNT = 000026            157#    628    1001    1081*   1089    1123    1779*   1959
PSENDA  005442           1087    1100    1108#
PSENDB  005674           1150    1157#
PSTIM   026111            930    1027    1106    3946#
PSW   = 177776             75#   1589*   2026*   2030*   3191*   3535*   4115*
PTIME   001424            425#   3144*   3147
PUSH  = 005746            138#
PUSH2 = 024646            139#
PWRCNT  001266 G           36#    349#   1713*   2635*   2918
PWRDN   015460            197     461    2627#   2633
PWRFAI  023702           2640    3672#
PWRFV   000024            197#    461*   2631*   2633*
PWRNUM  017224           1410    2916#
PWRUA   015552           2624    2641#   2678    2693
PWRUP   015506           2631    2633#
QOVRFL  022774           1261    1293    3589#
QTSTA   002272            563     565#
QTSTB   002304            561     566     568#
QTSTC   002316            559     569     571#
QUECNT  001332            376#    544     553
QUECTR  001027            254#    544*    548*    553*
QUETST  002214            548#    572     578     587     593     669     696     699    1074    1188    1348    1486    1524
QUE$  = 104401           1375    2537#
QUE.    006376           1279#   2536
RAND$ = 104420           2567#
RAND.   014514           2390#   2566
RANNUM  001200 G           35#    319#   2392    2412*
RANWRK  001202            320#   2391    2411#
RBUFEA= 000070            176#
RBUFPA= 000066            175#
RBUFSZ= 000072            177#
RBUFVA= 000064            174#
RDCUA = 000103             85#
RDFGIN= 000104             86#
RDPBA = 000102             84#
RDSERV= 000101             83#   4324
RDWHNI= 000022             87#   1762    2657
RELERR  024420           3556    3732#
RELMSG  024446           3540    3736#
RELOC   001315 G           36#    366#   1141    2741*   2800    4117*   4197*
RESIA   015302           2580    2591#
RESINT  015276            190    2590#   4103    4329
RESIV   000010            190#   4094*   4103*   4323*   4329*
RETRY   001331            375#   3457*   3563*   3564
RLOOP   014534           2394#   2410
```



```
RMODE   001311            362#    502*    571    1393    1396*   1418    1660    1748*   1752*   1775    1833    1836*   1840*
                         2661    3067
ROT     001035            262#   3332*   3334*
ROTBUF  016304           2761#   3335
ROTB1   016516           2797    2800#   2820
ROTB2   016554           2794    2809#
ROTB3   016600           2801    2803    2814#
ROTB4   016626           2819    2821#
ROTCHK  016740            509    2035    2847#
ROTCNT  001316            367#   1439*   2042*   2767*   2772*
ROTCT   001031            256#   1815*   1817*   1818*   1819*   1820*   1823*   1824    2788
ROTDIS  024555           2043    3751#
ROTENB  024505           2854    3743#
ROTI    001313            364#   1135    1813    2740*   2761    2852    4116*   4190*
ROTL    001314            365#    504*   2034*   2041*   2763    2775
ROTLOW  016652           1856    2798    2829#
ROTLO1  016606           2791    2817#
ROTLW1  016666           2824    2832#
ROTN    001337            382#   3334
ROTNUM  001320            368#   1439    2042    2772
ROTOFF  012760           2041#   3850
ROTON   012744           2034#   3846
ROTSIZ  001322            369#   1062
ROTXT   016650           2762    2768    2774    2776    2778    2786    2799    2808    2813    2815    2825#
RSTAT   001066            276#    693*    710     717*   1642*
RSTRT = 000062            173#    627    1079
RST04   014752           1685    1914    2344    2373    2413    2475#   2989    3154    3256    3787
RTIME   001414            421#   3065*   3077    3133    3137
RTIMX   001416            422#   3066*   3078    3138    3145*   3148
RTTK$ = 104423           1599    1622    2573#
RTTK.   014504           2380#   2572
RTT1    003136            713#   4099*
RUN     010710            538    1710#   3816
RUNC    011402           1774    1802#
RUND    011620           1805    1838#
RUNDIA  025365           3873#
RUNL    010700           1707#   3813
RUNPAS  020066           3098    3100    3103#
RUNRES  011134           1070    1755#   2758
RUNRFL  001326            372#    623    1069*   1750*   2636*
RUNSVC  002324            577#
RUNTYM  017656           1404    3045#   3132
SAVHI   001134            297#   2860    2863*   2865
SAVLO   001132            296#   2859    2861*   2864
SAVLOG  017506           2599    2708    2997#
SAV04   014740           1533    1648    1664    1901    2302    2352    2390    2467#   2947    3103    3202    3769
SBADR = 000052            168#   1539#
SEKND$  001412            420#   2987*   3084    3085    3151    3152
SEL     012462           1973#   3822
SETBUF  027252            473    4094#
SETCM1  027766           4182#
SETCOM  027[illegible]2   4155    4157    4175    4180#
SETXT   027430           4135    4158#
SETWBF  011644           1832    1845#   2814
SETWB1  011720           1855    1859#   2812
SETXT   030064           4122    4189    4200#
```

```
SHORT = ****** U          1    27    28    29   145   221   288   316   351   395   421   474   510
                        522   617   924  1021  1061  1077  1082  1092  1101  1110  1131  1391  1402
                       1409  1428  1550  1562  1569  1597  1610  1707  1714  1723  1756  1845  1856
                       1872  1997  2045  2084  2439  2500  2510  2581  2599  2603  2619  2627  2642
                       2664  2734  2791  2795  2817  2826  2857  2869  2994  3191  3196  3303  3339
                       3395  3573  3580  3602  3666  3680  3811  3851  3866  3935  4008  4010  4019
                       4104  4157  4158  4184  4319
SINPUT  010666        1700# 2122
SLDS    012514        1975  1984#
SLDSA   012514        1985#
SLDSB   012530        1721* 1973*  1979*  1989#
SLDSC   012542        1986  1991#
SLDSD   012546        1992# 1995
SLDSE   012552        1722* 1974*  1980*  1994#
SLDSL   012536        1987  1990#  1993
SMODE = 040000          93# 1893
SPBOT   027212         460   498   540  1387  2598  2618  2634  3534  4014#
SPCFLG  001010         239# 1656  1658*  1692*  1701*
SPOINT= 000024         156#  622
SPSAV   001152         304#  701*  1184  1200  1222  3190
SPSPEC  027252        2593  2608  4016#
SR      001176 G        35#  318#  463*   465   467*   506   758   802   922   975   979  1086  1595
                       3071  3142  3538
SRETRN  001056         272# 1659  1700*
SSP     027012        1894  4012#
SSR0  = 177572           78#  482*   497*   515*  1431*  2005*  2015  2020*  2581*  2630*  2644*  2646*  2730*
                       2746  2749*  3411*  3413*  4111*  4160*
SSR3  = 172516           79#
START   001534         229   231   333   344   413   459#  2757  4088
START2  002042         514#
START3  002062         513   520#
START4  002066         519   522#
STAT  = 000020         153#  597   603   608   618   693   694   697  1020*  1039  1046  1119  1121
                       1149  1152  1154*  1642  1781*  1782  1784  1788  1792  1795  1798  1947  1956
                       1989*  1994*  3219  3221  3223  3225  3227  3268*  3269
STAT1 = 000021         154#
STBFA   027414        4116#
STOP    001024         251#  581   586  1042  1050  1131  1147*  2771*  2773  2806*  3241
SUMARY  022753        1401  3584#
SVR0  = 000032         159#  1176  1540
SVR1  = 000034         160#
SVR2  = 000036         161#
SVR3  = 000040         162#
SVR4  = 000042         163#
SVR5  = 000044         164#
SVR6  = 000046         165#  622*   702  1221*  1634*  1644
SYSBAD  024145        2650  3702#
SYSCNT  001264         348# 1712*  2602*  2645  2647*  2648  2894
SYSERI  001310         361#  541*
SYSERR  023601        2611  3660#
SYSER1  023420        2621  3640#
SYSIZ   017052         510  2871#
SYSLIM  001350         392# 2648
SYSNUM  017140        1409  2892#
SYSSIZ  024465        2885  3739#
TABADR  001064         275#  754*   768   872*   959
```



```
TENPWR  014472        2355  2376#
TFLAG   001351         393# 3069  3097*  3099*  3101*  3114  3135  3140
THRS    026077        3079* 3080*  3081*  3943#
TIMX    026025        3290  3296  3935#
TIMX1   024230        3280  3711#
TITLE   022672         505  3573#
TKB     001164 G        35#  313#  1315  1326
TKS     001162 G        35#  312#   543*  1314*  1347*  1400*  2484  3420*
TMINS   026103        3082* 3083*  3944#
TOP12S  014622        1571  1573  2421#  4123  4313
TOTIM   026066         929  1026  1105  1406  3297  3941#
TO18S   014672        1845  2446#  2857  3541
TPB     001170 G        35#  315#  1392*  1522*  3304#
TPS     001166 G        35#  314#  1391*  1465*  1496*  3303#
TRAPX = 000024        2531# 2535#  2537#  2539#  2541#  2543#  2545#  2547#  2549#  2551#  2553#  2555#  2557#
                      2559# 2561#  2563#  2565#  2567#  2569#  2571#  2573#
TRCI    005700         192  1162#
TRCIA   005706        1164# 1167  4098#
TRCIB   005724        1168#
TRCIC   005710        1163  1165#
TRCPC   001070         277#  568   716   719*  1168*
TRCPSW  001072         278#  717  1169*
TRCV    000014         192#
TRPINT  015134         201  2518#
TRPLIM  015246        2523  2573#
TRPTAB  015176        2522  2532#
TRPV    000034         201#
TPP2  = 011000         738  2522  2530#
TSECS   026106        3084* 3085*  3945#
TSTMEM  030072        4182  4205#
TSTM1   030200        4227# 4243
TSTM2   030312        4254# 4271
TSTM3   030400        4275# 4293
TTPIV   000064         215#
TTSRV1  007436         455   567  1456  1487#  1520  1521
TTYBSY  001014         243#  562   729*   780*   983*  1038*  1108*
TTYBYT  001260         346# 1345*  1346
TTYLNK  001530         455# 1456*  1504*  1510*  1516*  1520*
TTYSRV  001526         215   454#  3305
TYPADR  001104         283# 1458*  1487  1500*
TYPCTR  001110         285#  565*  1523#
TYPDAT  004752         791   796   799   813   816   820   880  1008#
TYPE    007366         772   779   805   950   954   956   967   971   974  1448  1456#  2715  2719
                      3549  3562
TYPEB   007476        1488  1500#
TYPEC   007512        1503#
TYPED   007562        1516#
TYPEE   007556        1513  1515#  1519
TYPEL   007610        1502  1505  1511  1517  1522#
TYPEN   007350         763   927   936   944  1024  1033  1094  1101  1445#  3277  3293
TYPEQ   030072         147   728  1011  1260  1266  2491  2492  4018#
TYPE1   007352        1446# 1449
TYPLIM= 031222         147#  726  1009  1264
TYPQL = 001130         146#  147  2493  4018
TYPQUE  001007         238#  560   732*  1014*  1256  1268*
TYPQX   006172        1230#
```

```
TYPQ.    006232            1244#   2538    2540    2552    2562
TYPQ1    001100             281#   1258    1260*   1264    1266*   1267*   1269*   2491*
TYPQ1.   006244            1249#   2546
TYPQ2    001102             282#    726     728*    730     731*   1009    1011*   1012    1013*   1258    2492*
TYPQ2.   006156            1226#   2542    2544    2550
TYPQ3.   006056            1204#   2568
TYPQ4.   006010            1191#   2570
TYPRET   001106             284#   1457*   1497
TYPR1    001140             299#   1459*   1495
TYPR2    001142             300#   1460    1490
TYPR3    001144             301#
TYPR4    001146             302#
TYPR5    001150             303#
TYPSVC   003166             564     726#
TYPTAB   003250             738     741#
UIPAR0=  177640             110#   1581*
UIPAR1=  177642             111#   1582*   1619*
UIPAR2=  177644             112#   1583*   1615*
UIPAR7=  177656             113#   1584*
UIPDR0=  177600             106#   1577*
UIPDR1=  177602             107#   1578*
UIPDR2=  177604             108#   1579*
UIPDR7=  177616             109#   1580*
UMODE =  140000              92#   1589    1889
USP      027002            1890    4011#
WASADR=  000054             170#   1602*
WBFEA    001234             334#   1848    1851*   2811*   2823*   2830*   3318
WBFLG    001023             250#   1769*   2789*   3325
WBUF     001232             333#   1847    1849*   2779    2809*   2821*   2829*   3317
WBUFEA=  000076             179#   1553
WBUFPA=  000074             178#   1552    3316
WBUFRQ=  000100             180#
WBUFSZ=  000102             181#
WBUF12   001236             335#   1852*   1853*   1854    1863    2790    2792*   2793    2807*   2817*   2818    2831*   2833
WCASE    025006 G            33#   3775    3794#
WCASEE   025106 G            33#    414    3790    3806#
WTWHMI=  000222              88#   1764    2659
XCNT     001026             253#   1800*   1815    1821
XFLAG =  000005             152#    621*   3325    3327*
XFRFLG   001030             255#   3422*   3452*   3517
YES      001062             274#   2125    2165    2184*   2204*   2223*   2246    2248    2251*   2254    2261    2269*
ZERO     001222             328#
$GWBFC   001334             379#   1824*   2785*   2788*
$PASS    025756             766     932     939    3925#
.      = 030762              30#    186#    199     203#    212#    219     221#    228#    230#    397#    417#    418#    427#
                            1286   1561    1564    1629    1671    1709    1796    1799    1850    1869    2585    3405    3587
                            3813   3816#   3819#   3822#   3825#   3828#   3831    3834#   3837    3840#   3843    3846#   3850
                            3853   3856#   3859    3912#   4007#   4010#   4011#   4013#   4015#   4017#   4018#   4085#   4087#
                            4093#


. ABS.   032022      000
         000000      001


 ERRORS DETECTED:  0
```



```
 DEFAULT GLOBALS GENERATED:  0

 DXQABM,DXQABM/SOL/CRF:SYM=DXQABM
 RUN-TIME:  5 10 1 SECONDS
 RUN-TIME RATIO:  44/18=2.4
 CORE USED:  11K  (22 PAGES)
```

```
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;               IDENTIFICATION
                                ;
                                ;PRODUCT CODE:  MAINDEC-11-DXQAE-D-LA
                                ;
                                ;PRODUCT NAME:  DEC/X11 - SHORT MONITOR
                                ;
                                ;DATE:          JANUARY 1977
                                ;
                                ;MAINTAINER:    DIAGNOSTIC GROUP
                                ;
                                ;
                                ;
                                ;
                                ;COPYRIGHT 1973, 1974, 1975, 1976, 1977  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
```



```
                                        .LIST   SEQ
                                        .ENABL  AMA
                                        .TITLE  QAED  DEC/X11 SHORT MONITOR
          000000                        .ASECT
          000000                        .=0

                                        .GLOBL  LCOR12,HCOR12,DACNV,BDCNV,MODQ,CLOCKL,CLOCKP
                                        .GLOBL  KW11L,KW11P,HMS,HOURS,WCASE,WCASEE,CLOCK
                                        .GLOBL  OFFSET,HUNGCK,HNGTIM,LCLEAR,PCLEAR,CHKSUM
                                        .GLOBL  TKS,TKB,TPS,TPB,FILCNT,FILLER,SR,RANNUM,CAST
                                        .GLOBL  MODCNT,MODCTR,PASREL,RELOC,ALLBK,KTPRES,PWRCNT

                                ;******IF RUNNING WITH AN 11/60 AND ANY SYSTEM TYPE ERROR OCCURS
                                ;       (TRAP TO 4, ETC.) "LOGBUF" WILL CONTAIN
                                ;       THE CONTENTS OF SCRATCHPAD LOCATIONS FROM 0-177. SCRATCHPAD
                                ;       LOCATIONS 100-107 ARE PRINTED OUT.

                                ;SWITCH REGISTER OPTIONS:

                                        ;SR15=1  HALT MODULE AFTER ERROR.
                                        ;SR14=1  INHIBIT MODULE HALT AFTER 20 ERRORS.
                                        ;SR13=1  INHIBIT ERROR PRINT.
                                        ;SR12=1  ENABLE ENDPAS PRINTOUT.
                                        ;SR10=1  PRINT ALL DATA ERRORS IN BUFFER
                                        ;SR9=1   INHIBIT "RELOCATED TO " MESSAGE

                                ;MODULE STATUS BIT DEFINITIONS:

                                ;       BIT 15=1        IOMOD
                                ;       BIT 14=1        MODULE SELECTED
                                ;       BIT 13=1        MODULE DROPPED
                                ;       BIT 12=1        EXTENDED MODULE
                                ;       BIT 11=1        DESELECT IF PROGRAM IS RELOCATED
                                ;       BIT 10=1        DESELECT IF NOT ON AN EVEN 32K OFFSET
                                ;       BIT 09=1        NRKMOD
                                ;       ALL 0'S         SRKMOD
```

```
                                ;A FEW DEFINITIONS.
                                
000000                          R0      =%0     ;GENERAL PURPOSE REGISTER 0
000001                          R1      =%1     ;GENERAL PURPOSE REGISTER 1
000002                          R2      =%2     ;GENERAL PURPOSE REGISTER 2
000003                          R3      =%3     ;GENERAL PURPOSE REGISTER 3
000004                          R4      =%4     ;GENERAL PURPOSE REGISTER 4
000005                          R5      =%5     ;GENERAL PURPOSE REGISTER 5
000006                          R6      =%6     ;GENERAL PURPOSE REGISTER 6
000006                          SP      =%6     ;POINTER TO HARDWARE (SYSTEM DEFAULT) STACK
000007                          PC      =%7     ;PROGRAM COUNTER
                                
177776                          PS      =177776 ;PROCESSOR STATUS REGISTER
177776                          PSW     =177776 ;PROCESSOR STATUS WORD
177570                          HARDSR  =177570 ;HARDWARE SWITCH REGISTER
                                
177572                          SSR0    =177572 ;MEMORY MANAGEMENT REGISTER 0
172516                          SSR3    =172516 ;MEMORY MANAGEMENT REGISTER 3
                                
170200                          MAPREG  =170200 ;BASE ADDRESS OF UNIBUS MAP BOX (11/70)
```



```
076600                          MED     =76600  ;MAINTENANCE EXAMINE AND DEPOSIT INSTRUCTION (11/60)
000101                          RDSERV  =101    ;LOG SERVICE REG, ADR -- READ
000102                          RDPBA   =102    ;LOG PHYSICAL BUSS ADR, REG, ADR, -- READ
000103                          RDCUA   =103    ;LOG CURRENT MICRO ADR, REG, ADR, -- READ
000104                          RDFGIN  =104    ;LOG FLAG/INTERRUPT REG, ADR, -- READ
000022                          RDWHMI  =022    ;GEN, WHO AM I REG, ADR, -- READ
000222                          WTWHMI  =222    ;GEN, WHO AM I REG, ADR, -- WRITE
                                
177772                          PIRQRG  =177772 ;PROGRAMMED INTERRUPT REQUEST REGISTER
                                
140000                          UMODE   =140000 ;BIT CONFIGURATION TO INDICATE USER MODE IN PSW
040000                          SMODE   =40000  ;BIT CONFIGURATION TO INDICATE SUPERVISOR MODE IN PSW
                                
172300                          KIPDR0  =172300 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 0
172302                          KIPDR1  =172302 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 1
172304                          KIPDR2  =172304 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 2
172306                          KIPDR3  =172306 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 3
172316                          KIPDR7  =172316 ;KERNEL INSTRUCTION PAGE DESCRIPTOR REGISTER 7
172340                          KIPAR0  =172340 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 0
172342                          KIPAR1  =172342 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 1
172344                          KIPAR2  =172344 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 2
172346                          KIPAR3  =172346 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 3
172356                          KIPAR7  =172356 ;KERNEL INSTRUCTION PAGE   ADDRESS  REGISTER 7
                                
177600                          UIPDR0  =177600 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 0
177602                          UIPDR1  =177602 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 1
177604                          UIPDR2  =177604 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 2
177616                          UIPDR7  =177616 ;USER  INSTRUCTION PAGE DESCRIPTOR REGISTER 7
177640                          UIPAR0  =177640 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 0
177642                          UIPAR1  =177642 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 1
177644                          UIPAR2  =177644 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 2
177656                          UIPAR7  =177656 ;USER  INSTRUCTION PAGE   ADDRESS  REGISTER 7
```

```
                        PIRQ8   =IOT     ;MAKE THE BIT SIGNAL A TRAP THROUGH THE IOT VECTOR
                        OPEN    =0      ;USE OPEN TO DECLARE ALL VARIABLES AS 0 BEFORE STARTING
                        BIT15   =100000 ;BIT 15 DEFINITION
                        BIT14   =40000  ;BIT 14 DEFINITION
                        BIT13   =20000  ;BIT 13 DEFINITION
                        BIT12   =10000  ;BIT 12 DEFINITION
                        BIT11   =4000   ;BIT 11 DEFINITION
                        BIT10   =2000   ;BIT 10 DEFINITION
                        BIT9    =1000   ;BIT  9 DEFINITION
                        BIT8    =400    ;BIT  8 DEFINITION
                        BIT7    =200    ;BIT  7 DEFINITION
                        BIT6    =100    ;BIT  6 DEFINITION
                        BIT5    =40     ;BIT  5 DEFINITION
                        BIT4    =20     ;BIT  4 DEFINITION
                        BIT3    =10     ;BIT  3 DEFINITION
                        BIT2    =4      ;BIT  2 DEFINITION
                        BIT1    =2      ;BIT  1 DEFINITION
                        BIT0    =1      ;BIT  0 DEFINITION

                        PRTY7   =340    ;BIT CONFIGURATION FOR PRIORITY LEVEL 7 IN PSW
                        PRTY6   =300    ;BIT CONFIGURATION FOR PRIORITY LEVEL 6 IN PSW
                        PRTY5   =240    ;BIT CONFIGURATION FOR PRIORITY LEVEL 5 IN PSW
                        PRTY4   =200    ;BIT CONFIGURATION FOR PRIORITY LEVEL 4 IN PSW

                        PUSH    =005746 ;TST    -(SP)   ;;USED TO ADD A WORD TO  THE STACK
                        PUSH2   =024646 ;CMP    -(SP),-(SP)     ;;USED TO PUT TWO WORDS ON THE STACK
                        POPSP   =005726 ;TST    (SP)+   ;;USED TO DELETE A WORD FROM THE STACK
                        POPSP2  =022626 ;CMP    (SP)+,(SP)+;USED TO DELETE TWO WORDS FROM THE STACK

                        IE      =BIT6   ;DEFINITION OF INTERRUPT ENABLE BIT FOR ALL PDP-11 DEVICES
                        KBUFL   =32.    ;DEFINITION OF KEYBOARD BUFFER LENGTH (IN WORDS)
                        IOQL    =200.   ;DEFINITION OF I/O SERVICE QUEUE LENGTH (IN BYTES)
                        TYPQL   =300.   ;DEFINITION OF TYPING SERVICE QUE LENGTH (IN BYTES)
                        TYPLIM  =TYPEQ+TYPQL    ;DEFINITION OF LAST WORD IN TYPE QUEUE
                        IOQLIM  =IOQ+IOQL       ;DEFINITION OF LAST WORD IN I/O QUEUE
```



```
                        ;THE FOLLOWING DEFINITIONS ARE THE INDEX VALUES USED BY MODE 6 AND 7
                        ;MONITOR INSTRUCTIONS WHEN REFERRING TO LOCATIONS IN THE HEADER OF A MODULE

000005                  XFLAG   =5.    ;REFERENCE USED TO KEEP TRACK OF WRITE BUFFER USAGE
000020                  STAT    =16.   ;POINTS TO THE LOW BYTE OF THE MODULE STATUS WORD
000021                  STAT1   =17.   ;POINTS TO THE HIGH BYTE OF THE MODULE STATUS WORD
000022                  INIT    =18.   ;POINTS TO THE MODULE 'S STARTING ADDRESS
000024                  SPOINT  =20.   ;POINTS TO THE MODULE'S STACK POINTER
000026                  PSCNT   =22.   ;REFERENCES THE LOCATION KEEPING TRACK OF THE NUMBER OF PASSES
000030                  ERCNT   =24.   ;REFERENCES THE LOCATION KEEPING TRACK OF THE NUMBER OF ERRORS
000032                  SVR0    =26.   ;POINTS TO WHERE THE MODULE'S REGISTER 0 CONTENTS ARE STORED
000034                  SVR1    =28.   ;POINTS TO WHERE THE MODULE'S REGISTER 1 CONTENTS ARE STORED
000036                  SVR2    =30.   ;POINTS TO WHERE THE MODULE'S REGISTER 2 CONTENTS ARE STORED
000040                  SVR3    =32.   ;POINTS TO WHERE THE MODULE'S REGISTER 3 CONTENTS ARE STORED
000042                  SVR4    =34.   ;POINTS TO WHERE THE MODULE'S REGISTER 4 CONTENTS ARE STORED
000044                  SVR5    =36.   ;POINTS TO WHERE THE MODULE'S REGISTER 5 CONTENTS ARE STORED
000046                  SVR6    =38.   ;POINTS TO WHERE THE MODULE'S REGISTER 6 CONTENTS ARE STORED
000050                  CSRA    =40.   ;REFERENCES THE BASE ADDRESS OF THE MODULE'S DEVICES(CONTROL AND
000052                  ACSR    =42.   ;REFERENCES THE CONTENTS OF THE CONTROL AND STATUS REGISTER
000052                  SBADR   =42.   ;POINTS TO THA ADDRESS OF WHAT THE DATA SHOULD HAVE BEEN
000054                  ASTAT   =44.   ;POINTS TO ANYSTATUS REGISTERNOT THE SAME AS THE CSR
000054                  WASADR  =44.   ;POINTS TO THE ADDRESS OF WHAT THE DATAWAS
000056                  ASB     =46.   ;POINTS TO THEACTUAL EXPECTED DATA
000060                  AWAS    =48.   ;POINTS TO THE ACTUAL RECEIVED DATA
000062                  RSTRT   =50.   ;POINTS TO THE MODULE RESTART ADDRESS
000064                  RBUFVA  =52.   ;REFERENCES THE VIRTUAL ADDRESS OF THE READ BUFFER
000066                  RBUFPA  =54.   ;POINTS THE LOCATION FOR THE PHYSICAL ADDRESS OF THE READ BUFFER
000070                  RBUFEA  =56.   ;POINTS THE EXTENDED ADDRESS BITS OF THE READ BUFFER
000072                  RBUFSZ  =58.   ;POINTS THE SIZE OF THE READ BUFFER
000074                  WBUFPA  =60.   ;POINTS TO THE WRITE BUFFER PHYSICAL ADDRESS
000076                  WBUFEA  =62.   ;POINTS TO THE EXTENDED ADDRESS BITS OF THE WRITE BUFFER
000100                  WBUFRQ  =64.   ;REFERENCES THE AMOUNT OF WRITE BUFFER REQUESTED BY MODULE
000102                  WBUFSZ  =66.   ;REFERENCES THE SIZE OF THE WRITE BUFFER
```

```
                                ;LOAD TRAP AND VECTOR AREA

                        000000          .=0
000000  000000  000000          .WORD   0,0
000004  012516          BUSEV:  BUSERR                  ;BUS ERROR TRAP POINTER.
000006  000340                  PRTY7
000010  012524          RESIV:  RESINT                  ;RESERVED INSTRUCTION TRAP.
000012  000340                  PRTY7
000014  004772          TRCV:   TRCI                    ;TRACE TRAP POINTER.
000016  000000                  0
000020  005510          IOTV:   PIRQ,                   ;SERVICE PIRQ CALL
000022  000340                  PRTY7                   ;PREVENT LOWER LEVEL MODULE FROM COMING IN ON
                                                        ;TOP OF HIGHER LEVEL ONES
000024  012636          PWRFV:  PWRDN                   ;POWER FAIL POINTER.
000026  000340                  PRTY7
000030  000032          EMTV:   .+2                     ;EMT POINTER
000032  000000                  HALT
000034  012404          TRPV:   TRPINT                  ;TRAP POINTER.
000036  000000                  0
                        000040          .=40
000040  000000                  .WORD   OPEN            ;LOAD MEDIUM INDICATOR.
000042  000000          DDPPTR: .WORD   0               ;CHAIN MODE ONLY. POINTS TO DDPMON.
000044  000000                  .WORD   0
000046  012752                  LOGIC
000050  000000                  OPEN
000052  000000                  OPEN
000054  000000G                 CAST                    ;GLOBAL POINTER FOR CAST MODIFICATION
000056  000001                  1                       ;ALSO FOR CAST
                        000060          .=60
000060  005644          KYBDV:  KBSRVC                  ;KYBD INTERRUPT POINTER.
000062  000340                  PRTY7
000064  001406          TTPIV:  TTYSRV                  ;TELETYPE PRINTER SERVICE
000066  000200                  PRTY4
```



```
                                ;FROM 70 THROUGH 776 FILLED WITH .+2 AND HALT.




                                ;LOAD STARTING ADDRESSES

                        000200          .=200
000200  000137  001414          JMP     START           ;GO TO START OF MONITOR.
                        001000          .=1000
001000  000137  001414          JMP     START           ;GO TO START OF MONITOR.


001004  000000          FAKESR: OPEN                    ;LOCATION OF PSEUDO SWITCH REG, WHEN USED
```

```
                                ;VARIABLE DATA AREA CLEARED EVERY TIME RUN IS ISSUED OR ^C IS TYPED
                                ;
001006  000                     IOQUE:  .BYTE   OPEN            ;IOQUE AND TYPQUE MUST BE IN SAME WORD!!
001007  000                     TYPQUE: .BYTE   OPEN
001010  000                     SPCFLG: .BYTE   OPEN            ;SPCFLG AND DIRIND MUST BE IN SAME WORD!!
001011  000                     DIRIND: .BYTE   OPEN
001012  000                     BKQUE:  .BYTE   OPEN
001013  000                     BRAKE:  .BYTE   OPEN
001014  000                     TTYBSY: .BYTE   OPEN            ;TELETYPE BUSY FLAG, 0= NOT BUSY,
001015  000                     MODCNT: .BYTE   OPEN
001016  000                     MODCTR: .BYTE   OPEN
001017  000                     ERRIND: .BYTE   OPEN            ;0=ERROR, NOT 0=DATA ERROR,
001020  000                     FILCTR: .BYTE   OPEN
001021  000                     MDXCTR: .BYTE   OPEN            ;EXTENDED I/O MODULES NOT YET RUN IN CURRENT BANK
001022  000                     MDXCNT: .BYTE   OPEN            ;EXTENDED I/O MODULE COUNT
001023  000                     WBFLG:  .BYTE   OPEN            ;CURRENT VALUE OF XFLAG BEING CHECKED FOR
001024  000                     STOP:   .BYTE   OPEN            ;STOP MODULES TO RELOCATE PROGRAM
001025  000                     BKCNT:  .BYTE   OPEN            ;BACKGROUND MODULE COUNT
001026  000                     XCNT:   .BYTE   OPEN            ;RUNNABLE EXTENDED MODULE COUNT (TO CALCULATE ROTCT)
001027  000                     QUECTR: .BYTE   OPEN
001030  000                     XFRFLG: .BYTE   OPEN
001031  000                     ROTCT:  .BYTE   OPEN            ;VALUE USED FOR BUFFER ROTATION TIMEOUT (16 TIMES
                                                                ;XCNT OR 377, WHICHEVER IS SMALLER)
001032  000                     PASTOT: .BYTE   OPEN            ;ALTERNATE RELOCATION COUNTER= TOTAL ENDPASS
                                                                ;CALLS IN CURRENT BANK
001033  000                     PASREL: .BYTE   OPEN
001034  000                     BKTIMR: .BYTE   OPEN            ;BKMOD INSTR CTR= LOADED FROM BKTIME
001035  000                     ROT:    .BYTE   OPEN            ;WBUF ROTATION RATE CTR= LOADED FROM ROTN
                                        .EVEN
001036  000000                  KBPTR:  OPEN
001040  000000                  MODPTR: OPEN                    ;MODULE POINTER
001042  000000                  ADDRLO: OPEN
001044  000000                  ADDRHI: OPEN
001046  000000                  NUMBER: OPEN
001050  000000                  NUMBPA: OPEN
001052  000000                  NUMBEA: OPEN
001054  000000                  DSTADR: OPEN
001056  000000                  SRETRN: OPEN
001060  000000                  IOBKID: OPEN
001062  000000                  YES:    OPEN
001064  000000                  TABADR: OPEN
001066  000000                  RSTAT:  OPEN
001070  000000                  TRCPC:  OPEN
001072  000000                  TRCPSW: OPEN
001074  000000                  IOQ1:   OPEN                    ;I/O QUEUE POINTERS,
001076  000000                  IOQ2:   OPEN
001100  000000                  TYPQ1:  OPEN                    ;LOAD TYPE QUEUE POINTER
001102  000000                  TYPQ2:  OPEN                    ;UNLOAD TYPE QUEUE POINTER
001104  000000                  TYPADR: OPEN
001106  000000                  TYPRET: OPEN
001110  000000                  TYPCTR: OPEN                    ;USED TO MONITOR CONSOLE TELETYPE TO PREVENT HANGING
001112  000000                  OASCEA: OPEN                    ;USED TO HOLD EA BITS FOR OCTAL TO
                                                                ;ASCII CONVERSION
001114  000000                  DERRCT: OPEN
001116  000000                  SAVLO:  OPEN
001120  000000                  SAVHI:  OPEN
```



```
001122  000000                  MODADR: OPEN                    ;ADDRESS OF CURRENTLY EXECUTING (IOQ LEVEL) MODULE
001124  000000                  TYPR1:  OPEN                    ;SAVE LOCATIONS FOR TYPE ROUTINE REGISTERS
001126  000000                  TYPR2:  OPEN
001130  000000                  TYPR3:  OPEN
001132  000000                  TYPR4:  OPEN
001134  000000                  TYPR5:  OPEN
001136  000000                  SPSAV:  OPEN
001140  000000                  ERRPOS: OPEN
001142  000000                  CDFLG:  OPEN                    ;INDICATE RETURN TO CHECKDATA ROUTINE
001144  000000                  CCSRA:  OPEN
                                ;
                                ;
```

```
                        ;VARIABLE AND FIXED DATA AREA NOT CLEARED WHEN RUN OR ^C IS ISSUED

001146  177560          TKS:    177560
001150  177562          TKB:    177562
001152  177564          TPS:    177564          ;IF THIS AND NEXT LOC ARE CHANGED,
001154  177566          TPB:    177566          ;CHANGE LOC.S LPWK1 AND LPWK2 IF PRESENT
001156  177570          SR:     177570                  ;ADDRESS OF THE SWITCH REGISTER
001160  123456          RANNUM: 123456          ;RANDOM NUMBER IS CALC'ED AND PASSED HERE
001162  022333          RANWRK: 22333           ;ALSO USED BY RAND.
001164  000000          MEMVA:  OPEN
001166  000000          MEMPA:  OPEN
001170  000000          MEMEA:  OPEN
001172  000000          MEM12:  OPEN
001174  000000          MEMSBE: OPEN
001176  000000          MEMWAS: OPEN
001200  000000          LOCORE: OPEN                    ;STARTING ADDR OF FREE CORE (MULTIPLE OF 1000 OCTAL)
001202  000000          ZERO:   0                       ;LOCORE EA BITS
001204  000000          LCOR12: OPEN                    ;TOP 12 BITS OF LOW CORE ADDRESS
001206  000000          HCOR12: OPEN                    ;CONTAINS ST. ADDR OF HIGHEST BUFFER.
                                                        ;(TOP 12 BITS ONLY)
001210  000000          HCORSV: OPEN
001212  001414          WBUF:   START                   ;CONTAINS CURRENT WRITE BUFFER ADDR.
001214  000000          WBFEA:  OPEN                    ;WRITE BUFFER EA BITS (SHIFTED TO POS 4,5)
001216  000000          WBUF12: OPEN                    ;TOP 12 BITS OF WBUF ADDRESS IF USING ROTATION
001220  000000          OFFMX:  OPEN                    ;TOP ALLOWABLE PROGRAM OFFSET
001222  000000          OFFSET: OPEN                    ;CURRENT PROGRAM OFFSET
001224  000000          OFFSTD: OPEN                    ;DESTINATION OFFSET
001226  000000          OFFSTM: OPEN                    ;SAVE LOCATION FOR DESTINATION OFFSET
                                                        ;DURING MOVE
001230  000000          MOVBNK: OPEN                    ;BANK BEING MOVED (SOURCE ADDRESS)
001232  000000          MAPRET: OPEN                    ;RETURN ADDRESS FROM MVCODE ROUTINE
001234  000024          ERRLIM: 20.
001236  006000          MXWBF:  AMODNM-START/2&177000   ;MAX SIZE OF WRITE BUFFER, VARIES
                                                        ;IF BUFFER ROTATION IS ENABLED
001240  000000          TTYBYT: OPEN
001242  000000          CXRET:  OPEN                    ;CTRLX RETURN ADDRESS
001244  000000          SYSCNT: OPEN                    ;TOTAL NUMBER OF SYSTEM ERRORS
001246  000000          PWRCNT: OPEN                    ;TOTAL NUMBER OF POWER FAILS
001250  000000          BKPTR:  OPEN                    ;QUEUE POINTER INITIALIZATION FOR RUNNING BKMODS
001252  000000          KTPRES: OPEN                    ;0= INHIBITED OR NOT PRESENT (LEAVE AS FULL
                                                        ;WORD FOR EASY MODIFICATION AND TO GUARANTEE
                                                        ;THAT FILCNT AND FILLER ARE IN SAME WORD)
001254      014         FILCNT: .BYTE   12.             ;FILCNT AND FILLER MUST BE IN SAME WORD
001255      000         FILLER: .BYTE   0
001256      000         SYSERI: .BYTE   OPEN
001257      000         RMODE:  .BYTE   OPEN            ;RUN MODE INDICATOR
001260      000         CHN:    .BYTE   OPEN
001261      000         ROTI:   .BYTE   OPEN
001262      000         ROTL:   .BYTE   OPEN            ;WRITE BUFFER ROTATION LOCKED FLAG
001263      000         RELOC:  .BYTE   OPEN            ;RELOCATION ALLOWED FLAG
001264  000000          ROTCNT: OPEN                    ;ROTATION COUNTER USED WITH ROTL
001266  000100          ROTNUM: 100                     ;ROTATION CONSTANT USED WITH ROTCNT
001270  000400          ROTSIZ: 400                     ;ROTATE PROGRAM BY 4K
001272      000         FILLID: .BYTE   OPEN            ;FILL INDICATOR.
001273      000         MOVING: .BYTE   OPEN            ;CODE BEING RELOCATED
001274      000         RUNRFL: .BYTE   OPEN
```



```
001275      000         LOCK:   .BYTE   OPEN            ;PREVENT ROTATION OF CODE EXECUTION
001276      000         MVDOWN: .BYTE   OPEN
001277      000         RETRY:  .BYTE   OPEN            ;RETRY COUNT FOR CODE RELOCATION
001300      010         QUECNT: .BYTE   10              ;# OF TIMES TO CHECK IOQUE BEFORE CHECKING
                                                        ;TYPEQ WHEN BOTH HAVE REQUESTS PENDING
001301      000         MDL45:  .BYTE   OPEN            ;NON-ZERO IF CPU IS 11/45
001302      000         $GWBFC: .BYTE   OPEN            ;WRITE BUFFER ROTATION COUNT
001303      000         MEMERF: .BYTE   OPEN
001304      010         BKTIME: .BYTE   10
001305      003         ROTN:   .BYTE   3               ;# OF GWBUF CALLS (LESS ONE) NEEDED PER
                                                        ;ROTBUF CALL
001306      000         PPRES:  .BYTE   OPEN            ;PARITY PRESENT, 0=INHIBITED OR NONEXISTENT
001307      000         CNT1:   .BYTE   OPEN
001310      000         ATE:    .BYTE   OPEN
001311      000         MFLAG:  .BYTE   OPEN
001312      000         HNGFLG: .BYTE   OPEN            ;FLAG INDICATES CKHUNG ROUTINE IS BUSY
001313      000         MOVFLG: .BYTE   OPEN            ;FLAG USED IN MVCODE ROUTINE
001314      000         MSGFLG: .BYTE   OPEN            ;FLAG INDICATING MSGN OR MSGS CALL
001315      000         DATFLG: .BYTE   OPEN            ;FLAG INDICATING CKDATA OR DATACK CALL
001316      012         SYSLIM: .BYTE   10.             ;MAX. NUMBER OF SYSTEM ERRORS ALLOWED
001317      000         TFLAG:  .BYTE   OPEN            ;TIMER FLAG INDICATING ERROR OR EOP
001320      000         NSTOP:  .BYTE   OPEN            ;FLAG PREVENTING STOP FROM BEING SET
001321      001         CLOCK:  .BYTE   1               ;CLOCK FLAG SET FOR SHORT MONITOR
001322      000         ALLBK:  .BYTE   OPEN            ;WHEN NON 0 MEANS RUN ALL BKMODS BEFORE RELOCATING
        001324                  .EVEN
001324  000000G         CLOCKL: KW11L                   ;WILL BE 0 IF NO KW11-L CLOCK
001326  000000G         CLOCKP: KW11P                   ;WILL BE 0 IF NO KW11-P CLOCK

001330  001604          HNGTIM: 900.                    ;HUNG TIME LIMIT IN SECONDS
001332  000000G         ALCLR:  LCLEAR                  ;SUBROUTINE ADDRESS IN L CLOCK MODULE
001334  000000G         APCLR:  PCLEAR                  ;SUBROUTINE ADDR IN P CLOCK MODULE
                                                        ;WILL DROP ANY I/O MODULE INACTIVE FOR 15. MINS.
```

```
                                ;       *** DO NOT CHANGE THE SEQUENCE OF THE NEXT 9 WORDS *** !
                                ;REFER TO THE DOCUMENT FOR THE MONITOR CHECKSUM MODULE (BKBA)
                                ;FOR THE USE OF THE NEXT SIX WORDS, AND HOW THE TWO "OPEN" LOCATIONS
                                ;ARE DEFINED WHEN A NEW VERSION OF THE MONITOR IS RELEASED
001336  137577                  CHKL2:  137577          ;LOW CHKSUM FOR PROC. WITH NO RTT INST.
001340  002641                  CHKH2:  2641            ;HIGH CHKSUM
001342  001414                  CHKSUM: .WORD   START            ;BEGIN ADDR OF MONITOR PURE AREA
001344  015232                          .WORD   WCASEE           ;END ADDR OF MONITOR PURE AREA
001346  137607                  CHKL1:  137607          ;CONTAINS LOW CKSM OF MONITOR PURE AREA (PROC WITH RTT)
001350  002641                  CHKH1:  2641            ;CONTAINS HI CKSM OF MONITOR PURE AREA
        001353                          .ODD
001353  000003                  HOURS:  .BLKB   3                ;HOURS IN ASCII
001356  000000                  MINITS: OPEN                     ;MINUTES IN ASCII
001360  000000                  SEKNDS: OPEN                     ;SECONDS IN ASCII
                                        .EVEN
```



```
                                ;VARIABLE QUE CALL AREA- KEPT IN FRONT OF START SINCE IT IS IMPURE CODE
                                ;COMMON "QUE CALL" ROUTINE, DOES A DIRECT DISPATCH RATHER THAN
                                ;REQUEING DUE TO MULTIPLE POINTS CALLING THIS ROUTINE WHICH
                                ;MAY OVERLAP IF REQUEUED.
001362  012700  001372          COMQUE: MOV     #CADDR,R0
001366  000137  002500                  JMP     IOQSVN
001372  000000                  CADDR:  OPEN                     ;DESTINATION ADDR.
001374  000000                  CSTART: OPEN                     ;MODULE START ADDR. (0 FOR MONITOR).
                                ;CHECKDATA ERROR CALL
001376  104421                  CDERR:  DERRX$
001400  000000                  CDERR1: OPEN                     ;ADDRESS OF CKDATA CALL
001402  000137  007202                  JMP CDRET
                                ;TELETYPE ROUTINE PIRQ CALL
                                ;TTYLNK IS FREQUENTLY MODIFIED DURING EXECUTION
001406  000004                  TTYSRV: PIRQ$
001410  006426                  TTYLNK: TTSRV1
001412  000000                          0
```

```
                                        ;STARTUP SEQUENCE
                                        ;
001414  000005                 START:  RESET                   ;CLEAR THE WORLD.
001416  012706  016546                 MOV     #SPBOT,R6       ;SET UP STACK.
001422  012737  012636  000024         MOV     #PWRDN,PWRFV    ;SET UP POWER FAIL VECTOR.
001430  012746  000004                 MOV     #BUSEV,-(SP)    ;SAVE PRESENT BUS ERR TRAP
001434  012737  177570  001156         MOV     #HARDSR,SR      ;SET SR=HARDWARE SR
001442  012737  001456  000004         MOV     #3$,BUSEV       ;SET NEW BUS ERR TRAP
001450  005777  177502                 TST     @SR             ;IS IT HARDWARE SR?
001454  000404                         BR      4$              ;IF YES, RESTORE TRAP
001456  012737  001004  001156  3$:    MOV     #FAKESR,SR      ;ELSE SET UP SOFTWARE SR
001464  022626                         CMP     (SP)+,(SP)+             ;RESTORE STACK
001466  012637  000004          4$:    MOV     (SP)+,BUSEV     ;RESTORE BUS ERR TRAP
001472  012737  016234  001250         MOV     #MODQ-2,BKPTR   ;INITIALIZE TO START BKMODS WITH 1ST ONE
001500  005737  001200                 TST     LOCORE          ;DONE BUFFER SETUP?
001504  001002                         BNE     2$              ;BR IF YES.
001506  004537  016606                 JSR     R5,SETBUF       ;NO, DO IT.
001512  004737  012272          2$:    JSR     PC,CLRQUS       ;CLEAR QUEUES.
001516  105037  001257                 CLRB    RMODE
001522  112737  177777  001021         MOVB    #-1,MDXCTR      ;INITIALIZE COUNTER
001530  105037  001262                 CLRB    ROTL            ;MAKE SURE ROTATION IS NOT LOCKED
001534  104406  014062                 MSG$,TITLE              ;TYPE THE TITLE
001540  022737  001004  001156         CMP     #FAKESR,SR      ;DO WE HAVE A REAL SR?
001546  001002                         BNE     55$             ;BR IF YES
001550  104406  014756                 MSG$,NOSWR              ;ELSE TELL HIM WHERE IT IS
001554  004737  013262          55$:   JSR     PC,ROTCHK       ;CHECK IF ROTATE ALLOWED AND TYPE RANGE.
                                        ;
001560  105737  000041                 TSTB    @#41            ;ANY LOAD MEDIUM BITS SET ?
001564  001402                         BEQ     1$              ;NO, CONTINUE
001566  104406  014431                 MSG$,CLR40              ;" TO EXERCISE LOAD MEDIUM YOU MUST CLEAR LOC 40
001572  005037  001224          1$:    CLR     OFFSTD          ;MAKE SURE PROGRAM IS NOT PARTIALLY RELOCATED
001576  004737  014060                 JSR     PC,MVCODE
001602  004737  012272                 JSR     PC,CLRQUS       ;CLEAR QUEUES IN CASE MVCODE OVER WROTE THEM
001606  004737  012354                 JSR     PC,CLRVEC       ;INITIALIZE VECTOR AREA
001612  105737  001252                 TSTB    KTPRES          ;KT11 IN USE ?
001616  001002                         BNE     5$              ;YES, CONTINUE
001620  004737  015016                 JSR     PC,LOADWC       ;NO, LOAD WORST CASE PATTERN IN BUFFER
001624  104406  014122          5$:    MSG$,DOT                ;TYPE DOT.
001630  105737  001260                 TSTB    CHN             ;CHAIN MODE?
001634  001405                         BEQ     INPUT           ;NO- GO AWAIT KEYBOARD INPUT
001636  112777  000015  177172  6$:    MOVB    #15,@KBPTR      ;CLEAN OUT KEYBOARD BUFFER WITH A CR
001644  000137  007420                 JMP     RUN             ;YES=START RUNNING.
                                        ;
001650  012706  016546          INPUT:  MOV     #SPROT,R6       ;RESET STACK.
001654  105037  001256                 CLRB    SYSERI          ;CLEAR SYSTEM ERROR INDICATOR.
001660  012737  016306  001036         MOV     #KBUF,KBPTR     ;INITIALIZE KYBD BUFFER POINTER.
001666  052777  000100  177252         BIS     #IE,@TKS        ;ENABLE KYBD INTERRUPTS.
001674  113737  001300  001027         MOVB    QUECNT,QUECTR
                                        ;
```



```
                                        ;QUEUE CHECK.  NOTE THAT IF IOBKID IS NEGATIVE, NO TYPE QUE REQUESTS WILL BE SERVICED
                                        ;
001702  105337  001027          QUETST: DECB    QUECTR          ;COUNT TO PREVENT BREAK CALLS FROM LOCKING OUT TYPING
001706  003004                         BGT     1$              ;ALSO PREVENTS HIGH SPEED INT DEVICES FROM
                                        ;                       ;LOCKING OUT TYPE QUEUE AND RUN SERVICE
                                        ;                       ;STARTUP OF MODULES
                                        ;
001710  113737  001300  001027         MOVB    QUECNT,QUECTR   ;RESET QUE COUNTER
001716  000405                         BR      2$
001720  105737  001006          1$:    TSTB    IOQUE           ;IO QUE REQUEST PENDING?
001724  001402                         BEQ     2$
001726  000137  002424                 JMP     IOQSVC          ;GO SERVICE IF YES.
001732  005737  001060          2$:    TST     IOBKID          ;STARTING I/O MODULES?
001736  100422                         BMI     QTSTC           ;BR IF YES.
001740  105737  001007          3$:    TSTB    TYPQUE          ;TYPE REQUEST PENDING?
001744  001412                         BEQ     QTSTB           ;BR IF NOT.
001746  105737  001014                 TSTB    TTYBSY          ;TTY BUSY?
001752  001002                         BNE     QTSTA           ;BR IF YES.
001754  000137  002650                 JMP     TYPSVC          ;NO, GO SERVICE TYPE QUEUE.
001760  005337  001110          QTSTA:  DEC     TYPCTR          ;CHECK FOR TTY HUNG
001764  001002                         BNE     QTSTB
001766  000137  006426                 JMP     TTSRV1          ;UNHANG TTY
001772  005737  001070          QTSTB:  TST     TRCPC           ;BACKGROUND MODULE PENDING?
001776  001402                         BEQ     QTSTC
002000  000137  002622                 JMP     BKQSVC          ;JUMP IF YES
002004  105737  001257          QTSTC:  TSTB    RMODE           ;IN RUN MODE?
002010  001734                         BEQ     QUETST          ;BR IF NOT, GO CHECK AGAIN.
```

```
                                        ;RUN MODE SERVICE ROUTINE.
                                        ;NOTE THAT MODULE'S STACK POINTER IS REINITIALIZED BUT OTHER
                                        ;REGISTERS HAVE ANY PREVIOUS VALUES RESTORED AT START OR RESTART

002012  105737  001013          RUNSVC: TSTB    BRAKE           ;IS THE BRAKE ON?
002016  001331                          BNE     QUETST          ;BR IF YES, DO NOT INIT MORE MODULES.
002020  105737  001322                  TSTB    ALLBK           ;RUN ALL BKMODS BEFORE RELOCATING ?
002024  001407                          BEQ     12$             ;NO, CONTINUE
002026  105737  001024                  TSTB    STOP            ;WAITING TO RELOCATE ?
002032  001407                          BEQ     13$             ;NO, CONTINUE
002034  012737  040020  001060          MOV     #40020,IOBKID   ;START UP ONLY BKMOD'S
002042  000415                          BR      1$              ;CONTINUE

002044  105737  001024          12$:    TSTB    STOP            ;STOP SET?
002050  001314                          BNE     QUETST          ;YES, DON'T START ANY MORE MODULES
002052  105737  001016          13$:    TSTB    MODCTR          ;MODCTR =0?
002056  001007                          BNE     1$              ;BR IF NOT.
002060  105737  001260                  TSTB    CHN             ;YES, ARE WE IN CHAIN MODE?
002064  100004                          BPL     1$              ;BR IF NOT.
002066  012737  040020  001060          MOV     #40020,IOBKID   ;SWITCH IT TO BACKGROUND AND PREVENT LOOPING
002074  000702                          BR      QUETST          ;THRU BACKGROUND MODULE SERIES MORE THAN ONCE
002076  062737  000002  001040  1$:     ADD     #2,MODPTR       ;POINT TO NEXT MODULE.
002104  017700  176730                  MOV     @MODPTR,R0      ;MODULE ADDR TO R0.
002110  001474                          BEQ     7$              ;BR IF NO ADDR.
002112  016046  000020                  MOV     STAT(R0),-(SP)  ;GET MODULE'S STATUS
002116  042716  016000                  BIC     #BIT12!BIT11!BIT10,(SP) ;MASK SPECIAL INDICATORS
002122  022637  001060                  CMP     (SP)+,IOBKID    ;CORRECT MODULE TYPE TO RUN?
002126  001134                          BNE     11$             ;NO- BRANCH
002130  005737  001222                  TST     OFFSET          ;IF OFFSET 0, IGNORE BITS 10 +11
002134  001414                          BEQ     2$              ; CONTINUE
```



```
002136  032760  004000  000020          BIT     #BIT11,STAT(R0) ;BIT11 SET?
002144  001125                          BNE     11$             ;YES- DON'T RUN MODULE SINCE OFFSET NOT 0
002146  032737  001777  001222          BIT     #1777,OFFSET    ;OFFSET 32K MULTIPLE?

002154  001404                          BEQ     2$              ;YES- IGNORE BIT 10
002156  032760  002000  000020          BIT     #BIT10,STAT(R0) ;NO- BIT10 SET?
002164  001115                          BNE     11$             ;YES- DON'T RUN MODULE
002166  005737  001060          2$:     TST     IOBKID          ;BACKGROUND MODULE?



002172  100406                          BMI     3$              ;BR IF NOT.
002174  112737  177777  001013          MOVB    #-1,BRAKE       ;YES, APPLY BRAKE.
002202  013737  001040  001250          MOV     MODPTR,BKPTR    ;RECORD MODO ADDRESS OF CURRENT BKMOD
002210  032760  010000  000020  3$:     BIT     #BIT12,STAT(R0) ;EXTENDED I/O MODULE?
002216  001402                          BEQ     4$              ;NO- BRANCH
002220  105237  001022                  INCB    MDXCNT          ;YES, COUNT IT
002224  105060  000005          4$:     CLRB    XFLAG(R0)       ;CLEAR INDICATOR
002230  016060  000024  000046          MOV     SPOINT(R0),SVR6(R0) ;SET UP MODULE SP POINTER.
002236  105737  001274                  TSTB    RUNPFL
002242  001004                          BNE     5$
002244  016037  000022  001054  44$:    MOV     INIT(R0),DSTADR ;SET UP DESTINATION ADDR.
002252  000407                          BR      6$
002254  016037  000062  001054  5$:     MOV     RSTRT(R0),DSTADR
002262  005760  000026                  TST     PSCNT(R0)       ;HAS MODULE BEEN THRU START YET?
002266  001001                          BNE     6$              ;BR IF YES
002270  000765                          BR      44$             ;ELSE GO BACK
002272  105337  001016          6$:     DECB    MODCTR          ;DECR COUNT OF MODS INITED.
002276  000137  002514                  JMP     IOQSVE          ;GO TO START MODULE
002302  022737  040000  001060  7$:     CMP     #40000,IOBKID   ;FINISHED WITH SPECIAL BKMOD ?
002310  001004                          BNE     8$              ;BR IF NOT.
002312  012737  041000  001060          MOV     #41000,IOBKID   ;YES, SWITCH TO NBKMOD
002320  000434                          BR      10$             ;CONTINUE
002322  022737  041000  001060  8$:     CMP     #41000,IOBKID   ;FINISHED WITH NBKMODS ?
002330  001013                          BNE     9$              ;NO, CONTINUE
002332  012737  140000  001060          MOV     #140000,IOBKID  ;YES, SWITCH TO IOMOD.
002340  112737  177777  001021          MOVB    #-1,MDXCTR      ;INITIALLY LOCK MDXCTR



002346  105037  001022                  CLRB    MDXCNT          ;INITIALIZE COUNT



002352  105037  001320                  CLRB    NSTOP           ;ALLOW STOP FLAG TO BE SET
002356  000415                          BR      10$             ; CONTINUE
002360  005737  001060          9$:     TST     IOBKID          ;IF ALREADY BACKGROUND
002364  100012                          BPL     10$             ;DON'T RESET MDXCTR



002366  113737  001022  001021          MOVB    MDXCNT,MDXCTR   ;SETUP MDXCTR
```

```
   591  002374  012737  040020  001060          MOV     #40020,IOBKID   ;SWITCH TO BACKGROUND MODE
   592
   593
   594
   595
   596  002402  013737  001250  001040          MOV     BKPTR,MODPTR    ;CONTINUE WITH NEXT BKMOD
   597                                  ;       MOVB    #1,RUNRFL       ;SET RUN RESTART FLAG11/60          ************
   598  002410  000403                          BR      11$
   599  002412  012737  016234  001040  10$:    MOV     #MODQ-2,MODPTR  ;POINT TO MODULE TABLE START.
   600  002420  000137  001702          11$:    JMP     QUETST
```



```
   601                                  ;I/O QUE SERVICE ROUTINE.
   602                                  ;SERVICES PIRQ$,QUE$, AND BPEAK$ CALLS
   603                                  ;NOTE THAT PIRQ SERVICE CAN OCCUR ON TOP OF THIS ROUTINE- SO
   604                                  ;LEVEL 7 PRIORITY IS REQUIRED IF COUNT IS DECREMENTED AFTER IOQ2
   605                                  ;IS MOVED
   606
   607  002424  023727  001076  017116  IOQSVC: CMP     IOQ2,#IOQLIM    ;REACHED LIMIT OF QUEUE?
   608  002432  103403                          BLO     1$              ;BR IF NOT.
   609  002434  012737  016606  001076          MOV     #IOQ,IOQ2       ;RESET IOQ2.
   610  002442  105337  001006          1$:     DECB    IOQUE           ;DECREMENT REQUEST COUNT
   611  002446  017700  176424                  MOV     @IOQ2,R0        ;GET PC+2 OF CALL.
   612  002452  062737  000002  001076          ADD     #2,IOQ2         ;UPDATE IOQ2.
   613  002460  010001                          MOV     R0,R1           ;SAVE PC+2 OF CALL
   614  002462  014046                          MOV     -(R0),-(SP)     ;PUSH CALL ON STACK
   615  002464  022726  104407                  CMP     #BREAK$,(SP)+   ;BREAK CALL?
   616  002470  001002                          BNE     2$              ;NO, CONTINUE
   617  002472  000137  003136                  JMP     BREAK.          ;YES, GO SERVICE IT
   618  002476  005720                  2$:     TST     (R0)+
   619
   620  002500  012037  001054          IOQSVN: MOV     (R0)+,DSTADR    ;GET DESTINATION ADDR.
   621  002504  012000                          MOV     (R0)+,R0                ;GET MODULE ADDR. IS IT 0?
   622  002506  001002                          BNE     IOQSVE          ;BR IF NOT. IT'S A MODULE.
   623  002510  000177  176340                  JMP     @DSTADR         ;GO DO MONITOR FUNCTION.
   624  002514  116037  000020  001066  IOQSVE: MOVB    STAT(R0),RSTAT  ;GET RUN STATUS.
   625  002522  032760  020000  000020  IOQSVA: BIT     #BIT13,STAT(R0) ;MODULE STOPPED?
   626  002530  001402                          BEQ     1$              ;NO, CONTINUE
   627  002532  000137  001702                  JMP     QUETST          ;YES, FORGET IT
   628  002536  032760  040000  000020  1$:     BIT     #BIT14,STAT(R0) ;IS MODULE SELECTED ?
   629  002544  001002                          BNE     2$              ;YES, CONTINUE
   630  002546  000137  001702                  JMP     QUETST          ;NO, FORGET IT
   631  002552  010037  001122          2$:     MOV     R0,MODADR       ;SAVE R0, (MODULE ADDR).
   632  002556  010637  001136                  MOV     R6,SPSAV        ;SAVE MONITOR STACK TOO!.
   633  002562  062700  000050                  ADD     #SVR6+2,R0      ;RESTORE MODULE'S REGS.
   634  002566  014006                          MOV     -(R0),R6                ;STARTING WITH STACK POINTER.
   635  002570  014005  IOQSVD: MOV     -(R0),R5
   636  002572  014004                          MOV     -(R0),R4
   637  002574  014003                          MOV     -(R0),R3
   638  002576  014002                          MOV     -(R0),R2
   639  002600  014001                          MOV     -(R0),R1
   640  002602  014000                          MOV     -(R0),R0
   641  002604  013746  001066          IOQSVB: MOV     RSTAT,-(SP)     ;LOAD RUN STATUS.
   642  002610  105066  000001                  CLRB    1(SP)           ;KERNEL, 1ST REGISTER SET
   643  002614  013746  001054                  MOV     DSTADR,-(SP)    ;LOAD DESTINATION ADDR.
   644  002620  000006                  RTT1:   RTT                     ;GO TO DESTINATION.
```

```
                                        ;BACKGROUND QUEUE SERVICED HERE.
                                        ;
002622  013737  001070  001054  BKQSVC: MOV     TRCPC,DSTADR    ;SET UP DESTINATION ADDR.
002630  113737  001072  001066          MOVB    TRCPSW,RSTAT    ;SET UP RUN STATUS.
002636  017700  176176                  MOV     @MODPTR,R0      ;MODULE START ADDR TO R0.
002642  005037  001070                  CLR     TRCPC           ;CLEAR BK MODULE WAITING INDICATOR.
                                                                ;IF ERROR OCCURS, WILL STOP MODULE FROM FURTHER
                                                                ;EXECUTION UNTIL ERROR IS REPORTED
002646  000725                          BR      IOQSVA          ;GO GET GOING.
```



```
                                        ;TYPE QUE SERVICE ROUTINE.
                                        ;REMOVES PC+2 OF CALL FROM Q, PASSING IT IN R1
                                        ;
002650  023727  001102  017572  TYPSVC: CMP     TYPQ2,#TYPLIM   ;REACHED UPPER END OF QUEUE?
002656  103403                          BLO     1$              ;BR IF NOT.
002660  012737  017116  001102          MOV     #TYPEQ,TYPQ2    ;YES, RESET TYPQ2.
002666  105237  001014          1$:     INCB    TTYBSY          ;INDICATE TTY BUSY.
002672  017701  176204                  MOV     @TYPQ2,R1       ;GET PC+2 OF CALL.
002676  062737  000002  001102          ADD     #2,TYPQ2        ;UPDATE TYPQ2.
002704  105337  001007                  DECB    TYPQUE          ;DECREMENT REQUEST COUNT.
002710  010146                          MOV     R1,-(SP)                ;PUSH PC+2 OF CALL
002712  005741                          TST     -(R1)           ;POINT TO CALL.
002714  010137  001046                  MOV     R1,NUMBER       ;SAVE ADDRESS OF CALL
002720  011101                          MOV     (R1),R1         ;GET CALL
002722  006301                          ASL     R1              ;TIMES 2.
002724  016101  171726                  MOV     TYPTAB-TRP2-4(R1),R1    ;FORM SERVICE ADDR.
002730  000201                          RTS     R1              ;GO TO IT, RESTORE R1.
                                        ;
002732  004626          TYPTAB: .WORD   PASEND
002734  004440  003524  003516          .WORD   ENDSVC,ERRSVC,ERSVC1
002742  003110  000001  003506          .WORD   MSG,,1,ERSVC2,MSGN,,1,1,1,1,MSGS,,1,1,DERRX,,MSGD,
002750  003004  000001  000001
002756  000001  000001  002774
002764  000001  000001  003244
002772  003150
```

```
                                        ;MSGN, ROUTINE, SERVICES CALLS TO TYPE ASCII MESSAGES.

003774  112737  177777  001314  MSGS.:  MOVB    #-1,MSGFLG      ;SET MSGS CALL FLAG
003002  000402                          BR      MSCONT          ;CONTINUE
003004  105037  001314          MSGN.:  CLRB    MSGFLG          ;SET MSGN CALL FLAG

003010  012137  001064          MSCONT: MOV     (R1)+,TABADR    ;GET ASCII TABLE ADDR.
003014  012137  001374                  MOV     (R1)+,CSTART    ;MODULE ADDR TO CSTART.
003020  010137  001372                  MOV     R1,CADDR        ;RESUME ADDR TO CADDR.
003024  013701  001374                  MOV     CSTART,R1       ;MODULE ADDR TO R1.
003030  032777  020000  176120          BIT     #BIT13,@SR      ;INHIBIT ERROR PRINT?
003036  001033                          BNE     MSG1.           ;BR IF YES.
003040  105737  001314                  TSTB    MSGFLG          ;IS THIS A MSGS CALL ?
003044  100410                          BMI     1$              ;YES, CONTINUE
003046  004737  004314                  JSR     PC,ENDCOM       ;NO, DO COMMON STUFF.
003052  004537  006340                  JSR     R5,TYPEN        ;TYPE
003056  014127                          CRCR                    ;2 CARRAGE RETURNS
003060  015541                          AEND                    ;COMMON HEADER.
003062  015576                          $PASS                   ;PASS #.
003064  000000                          0
003066  013701  001064          1$:     MOV     TABADR,R1       ;TABLE ADDR TO R1.
003072  012146                  2$:     MOV     (R1)+,-(SP)     ;GET MESSAGE ADDR.
003074  022716  177777                  CMP     #-1,(SP)                ;TERMINATOR?
003100  001412                          BEQ     MSG1.           ;BR IF YES, DONE.
003102  004737  006356                  JSR     PC,TYPE         ;NO, TYPE MESSAGE.
003106  000771                          BR      2$              ;GO DO IT AGAIN.
```



```
                                        ;MSG$ CALL SERVICED HERE

003110  012146                  MSG.:   MOV     (R1)+,-(SP)     ;ASCII MESSAGE ADDR TO STACK.
003112  010137  001372                  MOV     R1,CADDR        ;RESUME ADDR TO CADDR.
003116  005037  001374                  CLR     CSTART          ;INDICATE MONITOR QUE CALLING.
003122  004737  006356                  JSR     PC,TYPE         ;GO TYPE DESIRED MESSAGE.
003126  105037  001014          MSG1.:  CLRB    TTYBSY
003132  000137  001362                  JMP     COMQUE          ;GO QUEUE UP TO RESUME.
```

```
                        ;BREAK ROUTINE, SERVICES BREAK CALL.

003136  012100          BREAK.: MOV     (R1)+,R0                ;GET MODULE ADDR.
003140  010137  001054          MOV     R1,DSTADR       ;GET DESTINATION ADDR.
003144  000137  002514          JMP     IOQSVE          ;QUE UP TO RESUME
```



```
                        ;MSGD$ SERVICE- TYPE DATA ERROR COUNT AND TRANSFER SIZE
                        ;AND HALT MODULE IF NECESSARY

003150  010137  001372  MSGD.:  MOV     R1,CADDR        ;SET UP RESUME ADDRESS
003154  004737  004376          JSR     PC,TYPDAT       ;GET MODULE ADDRESS
003160  012601                  MOV     (SP)+,R1
003162  010137  001374          MOV     R1,CSTART       ;SAVE ADDRESS
003166  004537  011206          JSR     R5,FILLNM       ;GET MODULE NAME
003172  016141                  AERNM+1
003174  004737  004376          JSR     PC,TYPDAT       ;SETUP ERROR COUNT (ASCII)
003200  004537  011744          JSR     R5,BDCNV
003204  016153                  AERCT
003206  004737  004376          JSR     PC,TYPDAT       ;SETUP TOTAL WORDS TRANSFERRED
003212  004537  011744          JSR     R5,BDCNV
003216  016205                  AERTOT
003220  032777  020000  175730  BIT     #BIT13,@SR      ;INHIBIT ERROR PRINTOUT?
003226  001004                  BNE     MSGD1.          ;YES- BRANCH
003230  012746  016140          MOV     #AERNM,-(SP)    ;NO- TYPE MESSAGE
003234  004737  006356          JSR     PC,TYPE
003240  000137  004222  MSGD1.: JMP     ERSVND          ;CHECK FOR HALTING MODULE, AND RETURN IF NOT
```

```
                                        ;DERRX, ROUTINE, SERVICES EXTENDED DATA ERROR CALLS FROM THE MONITOR
                                        ;CHECKDATA ROUTINE, TYPES ERROR INFORMATION AND RETURNS

003244  112737  000001  001017  DERRX,: MOVB    #1,ERRIND
003252  005721                          TST     (R1)+           ;POINT TO RETURN LOCATION
003254  010137  001372                  MOV     R1,CADDR        ;SETUP RESUME ADDRESS
003260  004737  004376                  JSR     PC,TYPDAT       ;GET MODULE ADDRESS
003264  012601                          MOV     (SP)+,R1
003266  010137  001374                  MOV     R1,CSTART       ;AND SAVE IT
003272  004737  004376                  JSR     PC,TYPDAT
003276  012637  001046                  MOV     (SP)+,NUMBER    ;SETUP ADDRESS OF CKDATA CALL
003302  004737  004314                  JSR     PC,ENDCOM
003306  012702  000005                  MOV     #5,R2
003312  004737  004376          DERRX1: JSR     PC,TYPDAT
003316  005302                          DEC     R2
003320  001374                          BNE     DERRX1
003322  016146  000030                  MOV     ERCNT(R1),-(SP) ;ERROR COUNT TO STACK
003326  004537  011744                  JSR     R5,BDCNV        ;CONVERT ERROR COUNT TO DECIMAL
003332  016052                          AERNMB
003334  005046                          CLR     -(SP)
003336  005766  000002                  TST     2(SP)
003342  001402                          BEQ     1$
003344  012716  000060                  MOV     #60,(SP)        ;CSRA EA BITS
003350  004537  011524          1$:     JSR     R5,OACNVX       ;CSR ADDRESS- ASCII
003354  015722                          ADTE6
003356  012603                          MOV     (SP)+,R3        ;SAVE TABLE ADDRESS
003360  012637  001140                  MOV     (SP)+,ERRPOS    ;SAVE OFFSET COUNT
003364  006337  001140                  ASL     ERRPOS
003370  012346                          MOV     (R3)+,-(SP)     ;SETUP READ BUFFER PA
003372  012346                          MOV     (R3)+,-(SP)     ;GET EA BITS
003374  063766  001140  000002          ADD     ERRPOS,2(SP)
003402  103002                          BCC     2$
003404  062716  000020                  ADD     #20,(SP)
003410  005723                  2$:     TST     (R3)+           ;SKIP SIZE
003412  004537  011524                  JSR     R5,OACNVX       ;GET PA OF RDADR
003416  016002                          ADTE4
003420  012346                          MOV     (3)+,-(SP)      ;GET WRITE PA
003422  012346                          MOV     (3)+,-(SP)      ;GET EA BITS
003424  063766  001140  000002          ADD     ERRPOS,2(SP)
003432  103002                          BCC     3$
003434  062716  000020                  ADD     #20,(SP)
003440  004537  011524          3$:     JSR     R5,OACNVX       ;GET ASCII OF WRADR
003444  015765                          ADTE5
003446  004537  011512                  JSR     R5,OACNV        ;CONVERT S/B TO ASCII
003452  015735                          ADTE3
003454  004537  011512                  JSR     R5,OACNV        ;CONVERT WAS TO ASCII
003460  015750                          ADTE2
003462  013746  001140                  MOV     ERRPOS,-(SP)    ;ERROR POSITION IN BUFFER
003466  000241                          CLC                     ;CORRECT BACK FROM BYTE TO WORD PTR
003470  006216                          ASR     (SP)
003472  005216                          INC     (SP)            ;MAKE 1ST WORD #1
003474  004537  011744                  JSR     R5,BDCNV        ;GET DECIMAL
003500  016034                          ADTE7
003502  000137  004014                  JMP     ERSVX           ;GO TO COMMON ROUTINE
```



```
                                        ;ERROR CALLS ARE SERVICED HERE.
                                        ;R1 AT ENTRY CONTAINS PC+2 OF CALL

003506  112737  000003  001017  ERSVC2: MOVB    #3,ERRIND       ;INDICATE ERRORN CALL.
003514  000406                          BR      ERSVCA
003516  105037  001017          ERSVC1: CLRB    ERRIND          ;INDICATE DATA ERROR.
003522  000403                          BR      ERSVCA
003524  112737  000002  001017  ERRSVC: MOVB    #2,ERRIND       ;INDICATE "NORMAL" ERROR.
003532  012137  001374          ERSVCA: MOV     (R1)+,CSTART    ;SAVE START ADDR OF MODULE.
003536  122737  000003  001017          CMPB    #3,ERRIND               ;ERRORN CALL?
003544  001002                          BNE     1$              ;BR IF NOT
003546  012137  001064                  MOV     (R1)+,TABADR    ;SAVE TABLE ADDR.
003552  010137  001372          1$:     MOV     R1,CADDR        ;RESUME ADDR TO PSEUDB.
003556  013701  001374                  MOV     CSTART,R1       ;GET BACK START ADDR.
003562  004737  004314                  JSR     PC,ENDCOM       ;DO COMMON STUFF.
003566  005261  000030                  INC     ERCNT(R1)       ;INCREMENT MODULE'S ERROR COUNT.
003572  001002                          BNE     ERSVCB          ;BR IF RESULT NOT 0.
003574  005161  000030                  COM     ERCNT(R1)       ;RESET COUNT TO -1.
003600  012702  000005          ERSVCB: MOV     #5,R2           ;GET TYPE DATA FROM QUEUE TO STACK.
003604  004737  004376          ERSVCC: JSR     PC,TYPDAT       ;DO IT.
003610  005302                          DEC     R2              ;DONE?
003612  001374                          BNE     ERSVCC          ;BR IF NOT.
003614  016146  000030                  MOV     ERCNT(R1),-(SP) ;ERROR COUNT TO STACK.
003620  004537  011744                  JSR     R5,BDCNV        ;CONVERT ERROR COUNT TO DECIMAL.
003624  016052                          AERNMB
003626  105737  001017                  TSTB    ERRIND          ;DATA ERROR?
003632  001047                          BNE     2$              ;BR IF NOT.
003634  004537  011512                  JSR     R5,OACNV        ;CONVERT WAS TO ASCII.
003640  015750                          ADTE2
003642  004537  011512                  JSR     R5,OACNV        ;CONVERT S/B TO ASCII,RESET TYPE2.
003646  015735                          ADTE3
003650  012637  001046                  MOV     (SP)+,NUMBER    ;GET ADDR OF LOCATION READ
003654  104413  001046                  GETPAS,NUMBER
003660  013746  001050                  MOV     NUMBPA,-(SP)
003664  013746  001052                  MOV     NUMBEA,-(SP)
003670  004537  011524                  JSR     R5,OACNVX       ;CONVERT WASADR TO ASCII.
003674  016002                          ADTE4
003676  012637  001046                  MOV     (SP)+,NUMBER
003702  104413  001046                  GETPAS,NUMBER
003706  013746  001050                  MOV     NUMBPA,-(SP)
003712  013746  001052                  MOV     NUMBEA,-(SP)
003716  004537  011524                  JSR     R5,OACNVX       ;CONVERT SBADR TO ASCII.
003722  015765                          ADTE5
003724  005046                          CLR     -(SP)           ;0 ONTO STACK
003726  016637  000002  001144          MOV     2(SP),CCSRA     ;SAVE CSR ADDRESS
003734  001402                          BEQ     1$              ;IF 0, BRANCH
003736  012716  000060                  MOV     #60,(SP)        ;SETUP EA BITS
003742  004537  011524          1$:     JSR     R5,OACNVX       ;CONVERT CSR ADDR TO ASCII.
003746  015722                          ADTE6
003750  000421                          BR      ERSVX
003752  022627                  2$:     CMP     (SP)+,(SP)+             ;SKIP WAS AND S/B.
003756  00 37                          AS,OACNV                ;CONVERT STAT REG CONTENTS TO ASCII.
003760  0157 4                         ASTAT
003762  004537  011512                  JSR     R5,OACNV        ;CONVERT CSR CONTENTS TO ASCII.
003766  015667                          ACSRC
003770  005046                          CLR     -(SP)           ;0 ONTO STACK
```

```
 848  003772  016637  000002  001144          MOV     2(SP),CCSRA     ;SAVE CSR ADDRESS
 849  004000  001402                          BEQ     3$              ;BRANCH IF ADDRESS 0
 850  004002  012716  000060                  MOV     #60,(SP)
 851  004006  004537  011524          3$:     JSR     R5,OACNVX       ;CONVERT CSR ADDR TO ASCII.
 852  004012  015653                          ACSRAC
 853  004014  032777  020000  175134  ERSVX:  BIT     #BIT13,@SR      ;INHIBIT ERROR PRINT?
 854  004022  001077                          BNE     ERSVND          ;BR IF YES.
 855  004024  004537  006340          7$:     JSR     R5,TYPEN        ;TYPE
 856  004030  014127                          CRCR                    ;2 CARRAGE RETURNS
 857  004032  015541                          AEND                    ;COMMON HEADER.
 858  004034  015576                          @PASS                   ;PASS #.
 859  004036  016044                          ERRNMB                  ;ERROR #.
 860  004040  000000                          0
 861  004042  122737  000002  001017  8$:     CMPB    #2,ERRIND       ;DATA ERROR?
 862  004050  003406                          BLE     1$              ;BR IF NOT.
 863  004052  004537  006340                  JSR     R5,TYPEN        ;YES, TYPE
 864  004056  016012                          ADTE4A                  ;DATA ERROR VALUES
 865  004060  015714                          ADTERR
 866  004062  000000                          0
 867  004064  000404                          BR      2$
 868  004066  012746  015645          1$:     MOV     #AERROR,-(SP)   ;TYPE ERROR MESSAGE.
 869  004072  004737  006356                  JSR     PC,TYPE
 870  004076  122737  000001  001017  2$:     CMPB    #1,ERRIND       ;EXTENDED DATA ERROR?
 871  004104  001004                          BNE     3$              ;NO= BRANCH
 872  004106  012746  016026                  MOV     #ADTEX,-(SP)    ;YES= OUTPUT POSITION IN BUFFER
 873  004112  004737  006356                  JSR     PC,TYPE
 874  004116  012746  014155          3$:     MOV     #ACRLF,-(SP)
 875  004122  004737  006356                  JSR     PC,TYPE
 876  004126  122737  000003  001017          CMPB    #3,ERRIND       ;ERRORN CALL?
 877  004134  001032                          BNE     ERSVND          ;BR IF NOT.
 878  004136  013702  001064                  MOV     TABADR,R2       ;TABLE ADDR TO R2.
 879  004142  012703  000010          4$:     MOV     #8.,R3          ;WILL TYPE 8 VALUES PER LINE.
 880  004146  022712  177777          5$:     CMP     #-1,(2)         ;TERMINATOR?
 881  004152  001417                          BEQ     6$              ;BR IF YES. DONE.
 882  004154  013246                          MOV     @(2)+,-(SP)     ;PUT VALUE IN STACK.
 883  004156  004537  011512                  JSR     R5,OACNV        ;CONVERT IT TO OCTAL.
 884  004162  015515                          HOLD6                   ;AND PUT IT HERE
 885  004164  012746  015515                  MOV     #HOLD6,-(SP)    ;TYPE IT.
 886  004170  004737  006356                  JSR     PC,TYPE
 887  004174  005303                          DEC     R3              ;DONE 8 PER LINE?
 888  004176  001363                          BNE     5$              ;BR IF NOT.
 889  004200  012746  014155                  MOV     #ACRLF,-(SP)    ;OUTPUT CRLF.
 890  004204  004737  006356                  JSR     PC,TYPE
 891  004210  000754                          BR      4$              ;GO FOR MORE.
 892  004212  012746  014155          6$:     MOV     #ACRLF,-(SP)    ;OUTPUT CRLF.
 893  004216  004737  006356                  JSR     PC,TYPE
 894  004222  005777  174730          ERSVND: TST     @SR             ;HALT MODULE ON ERROR?
 895  004226  100410                          BMI     1$              ;BR IF YES.
 896  004230  023761  001234  000030          CMP     EPRLIM,EPCNT(R1)        ;ERROR COUNT 20 OR GREATER?
 897  004236  003010                          BGT     2$              ;BR IF NOT, CONTINUE MODULE EXECUTION.
 898  004240  032777  040000  174710          BIT     #BIT14,@SR      ;YES, HALT MODULE AFTER 20 ERRORS?
 899  004246  001004                          BNE     2$              ;NO= SKIP
 900  004250  005037  001144          1$:     CLR     CCSRA           ;CLEAR CCSRA
 901  004254  000137  004446                  JMP     ENDSVA          ;YES, GO HALT MODULE
 902  004260  105037  001014          2$:     CLRB    TTYBSY          ;QUE TO RESUME
 903  004264  005737  001144                  TST     CCSRA           ;IF NOT SETUP, DON'T RESTORE INTO MODULE
```



```
 904  004270  001407                          BEQ     3$
 905  004272  013700  001374                  MOV     CSTART,R0
 906  004276  013760  001144  000050          MOV     CCSRA,CSRA(R0)  ;RESTORE CSRA IN MODULE
 907  004304  005037  001144                  CLR     CCSRA
 908  004310  000137  001362          3$:     JMP     COMQUE
```

```
   909  004314  104413  001046          ENDCOM: GETPA$,NUMBER           ;NUMBER CONTAINS ADDRESS OF CALL
   910  004320  013746  001050                  MOV     NUMBPA,-(SP)
   911  004324  013746  001052                  MOV     NUMBEA,-(SP)
   912  004330  013746  001046                  MOV     NUMBER,-(SP)
   913  004334  160116                          SUB     R1,(SP)         ;COMPUTE ASSEMBLY PC.
   914  004336  004537  011512                  JSR     R5,OACNV        ;CONVERT ASSEMBLY PC TO ASCII.
   915  004342  015566                          AEND2
   916  004344  004537  011524                  JSR     R5,OACNVX       ;CONVERT PC TO ASCII.
   917  004350  015553                          AEND1
   918  004352  004537  011206                  JSR     R5,FILLNM       ;LET'S GET MODULE NAME.
   919  004356  015541                          AEND                    ;STUFF AT AEND.
   920  004360  016146  000026                  MOV     PSCNT(R1),-(SP) ;GET PASS COUNT.
   921  004364  005216                          INC     (SP)            ;UP IT 1.
   922  004366  004537  011744                  JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
   923  004372  015606                          APASS
   924  004374  000207                          RTS     PC              ;LET'S GET OUT.
```



```
   925                                  ;TYPDAT ROUTINE LOADS QUEUED DATA ONTO STACK.
   926
   927  004376  011646                  TYPDAT: MOV     (SP),-(SP)      ;SAVE EXIT ON STACK AGAIN.
   928  004400  023727  001102  017572          CMP     TYPQ2,#TYPLIM   ;REACHED END OF QUEUE?
   929  004406  103403                          BLO     1$              ;BR IF NOT.
   930  004410  012737  017116  001102          MOV     #TYPEQ,TYPQ2    ;YES, POINT TO START OF QUEUE.
   931  004416  017766  174460  000002  1$:     MOV     @TYPQ2,2(SP)    ;QUEUE DATA TO STACK.
   932  004424  062737  000002  001102          ADD     #2,TYPQ2        ;UPDATE QUEUE POINTER.
   933  004432  105337  001007                  DECB    TYPQUE          ;DECREMENT COUNT.
   934  004436  000207                          RTS     PC              ;EXIT, LEAVE DATA IN STACK.
```

```
                                        ;END CALL SERVICED HERE.
                                        
004440  011101                  ENDSVC: MOV     (R1),R1         ;GET START ADDR.
004442  004737  004314                  JSR     PC,ENDCOM       ;DO COMMON STUFF.
004446  052761  020000  000020  ENDSVA: BIS     #BIT13,STAT(R1) ;SET STOP BIT IN MODULE STAT.
                                        
004454  004537  006340          1$:     JSR     R5,TYPEN        ;TYPE
004460  014127                          CRCR                    ;2 CARRAGE RETURNS
004462  015541                          AEND                    ;COMMON HEADER.
004464  015615                          MODEND                  ;END MESSAGE.
004466  000000                          0
004470  105037  001014          2$:     CLRB    TTYBSY          ;CLEAR TTY BUSY INDICATOR.
004474  005761  000020          ENDSVE: TST     STAT(R1)                ;BACKGROUND MODULE?
004500  100410                          BMI     1$              ;BR IF NOT.
004502  105037  001013                  CLRB    BRAKE           ;RELEASE BRAKE.
004506  105737  001024                  TSTB    STOP            ;STOP MODULES FOR RELOCATION?
004512  001403                          BEQ     1$              ;NO, CONTINUE
004514  105737  001322                  TSTB    ALLBK           ;RUN ALL BKMODS BEFORE RELOCATION ?
004520  001413                          BEQ     ENDSVG          ;NO, CONTINUE
004522  032761  010000  000020  1$:     BIT     #BIT12,STAT(R1) ;EXTENDED I/O MOD?
004530  001402                          BEQ     6$              ;NO= BRANCH
004532  105337  001022                  DECB    MDXCNT          ;YES, DEC EXTENDED MOD COUNT
004536  105337  001015          6$:     DECB    MODCNT          ;DECR COUNT OF MODULES RUNNING.
004542  105737  001024                  TSTB    STOP            ;RELOCATE CODE?
004546  001420                          BEQ     ENDSVF          ;NO= BRANCH
004550  123737  001015  001025  ENDSVG: CMPB    MODCNT,BKCNT    ;ALL I/O MODULES OFF?
004556  003021                          BGT     ENDSVD          ;NO= BRANCH
004560  105737  001322                  TSTB    ALLBK           ;RUN ALL BKMOD'S BEFORE RELOCATING ?
004564  001404                          BEQ     1$              ;NO, CONTINUE
004566  105737  001015                  TSTB    MODCNT          ;ALL MODULES DONE ?
004572  001404                          BEQ     ENDSVB          ;YES, GO RELOCATE
004574  000412                          BR      ENDSVD          ;NO, GO RUN THE REST OF THE MODULES
004576  105737  001013          1$:     TSTB    BRAKE           ;YES= BKMOD OFF?
004602  001007                          BNE     ENDSVD          ;NO= BRANCH (STOP BKMOD CLEANLY)
004604                          ENDSVB:
004604  000137  007476                  JMP     RUNRES          ;GO STARTUP AGAIN
004610  105737  001015          ENDSVF: TSTB    MODCNT          ;ANY MODULES RUNNING?
004614  001002                          BNE     ENDSVD          ;YES= BRANCH
004616  000137  006142                  JMP     CTRLCB          ;COUNT 0, TERMINATE RUN MODE.
004622  000137  001702          ENDSVD: JMP     QUETST          ;GO BACK TO SERVICE QUEUES.
```



```
                                        ;PASEND ROUTINE, TYPES END OF PASS MESSAGE.
                                        ;BACKGROUND MODULES ARE NOT ALLOWED TO MAKE MULTIPLE PASSES.
                                        ;IN CHAIN MODE NO MODULE ALLOWED TO MAKE MULTIPLE PASSES.
                                        
004626  011101                  PASEND: MOV     (R1),R1         ;LOAD R1 WITH MODULE ADDRESS
004630  016137  000062  001372          MOV     RSTRT(R1),CADDR ;SAVE RESTART ADDRESS
004636  010137  001374                  MOV     R1,CSTART       ;SAVE MODULE ADDRESS
004642  005261  000026                  INC     PSCNT(R1)       ;INCREMENT PASS COUNT.
004646  032777  010000  174302  1$:     BIT     #BIT12,@#R      ;ENDPAS PRINTOUT?
004654  001417                          BEQ     PSENDA          ;BR IF NOT.
004656  004737  004314                  JSR     PC,ENDCOM       ;DO COMMON STUFF.
004662  016146  000026                  MOV     PSCNT(R1),-(SP) ;NOW GET IT.
004666  004537  011744                  JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
004672  015636                          BPSCNT                  ;STUFF IT AT BPSCNT.
004674  004537  006340                  JSR     R5,TYPEN        ;TYPE
004700  014127                          CRCR                    ;2 CARRAGE RETURNS
004702  015541                          AEND                    ;COMMON HEADER.
004704  015627                          APSEND                  ;ENDPAS AND COUNT.
004706  014125                          CR
004710  000000                          0
004712  000400                          BR      PSENDA          ;CONTINUE
004714  105037  001014          PSENDA: CLRB    TTYBSY          ;CLEAR TTY BUSY INDICATOR.
004720  105737  001260                  TSTB    CHN             ;IN CHAIN MODE?
004724  100663                          BMI     ENDSVE          ;BR IF YES TO END MODULE EXECUTION
004726  005761  000020          2$:     TST     STAT(R1)                ;BACKGROUND MODULE?
004732  100415                          BMI     PSENDB          ;BR IF NOT.
004734  105037  001013                  CLRB    BRAKE           ;RELEASE BRAKE.
004740  032761  001000  000020          BIT     #BIT9,STAT(R1)  ;IS IT AN NBKMOD ?
004746  001725                          BEQ     ENDSVD          ;NO, CONTINUE
004750  052761  020000  000020          BIS     #BIT13,STAT(R1) ;YES, SET IT'S STOP BIT
004756  105337  001025                  DECB    BKCNT           ;ONE LESS BKMOD RUNNING
004762  000137  004474                  JMP     ENDSVE          ;GO TAKE CARE OF OTHER STUFF
004766  000137  001362          PSENDB: JMP     COMQUE          ;GO TO COMMON QUE CALL.
```

```
                                ;TRACE TRAP ENTERS HERE.
                        
004772  105337  001034        TRCI:   DECB    BKTIMR
004776  100401                        BMI     TRCIC
005000  000006                TRCIA:  RTT
005002  113737  001304  001034TRCIC:  MOVB    BKTIME,BKTIMR
005010  005737  001006                TST     IOQUE           ;I/O OR TYPE QUE WAITING?
005014  001771                        BEQ     TRCIA           ;NO, EXIT.
005016  012637  001070        TRCIB:  MOV     (SP)+,TRCPC     ;SAVE MOD'S PC.
005022  012637  001072                MOV     (SP)+,TRCPSW    ;SAVE MOD'S PSW.
005026  000401                        BR      EXIT1.
```



```
                                ;EXIT CALL ENTERS HERE.
                        
005030  022626                  EXIT.:  CMP     (SP)+,(SP)+
005032  010046                  EXIT1.: MOV     R0,-(SP)                ;SAVE R0 IN STACK.
005034  013700  001122                  MOV     MODADR,R0       ;MODULE ADDR TO R0.
005040  062700  000032                  ADD     #SVR0,R0        ;POINT TO MOD'S REG SAVE AREA.
005044  012620                          MOV     (SP)+,(R0)+     ;SAVE R0, (FROM STACK).
005046  010120                          MOV     R1,(R0)+                ;SAVE REMAINING REGS.
005050  010220                          MOV     R2,(R0)+
005052  010320                          MOV     R3,(R0)+
005054  010420                          MOV     R4,(R0)+
005056  010520                          MOV     R5,(R0)+
005060  010610                          MOV     R6,(R0)         ;SAVE MODULE STACK POINTER.
005062  013706  001136                  MOV     SPSAV,R6        ;RESTORE MONITOR STACK.
005066  005046                  EXIT2.: CLR     -(SP)           ;CLEAR PSW ON STACK
005070  012746  005076                  MOV     #1$,-(SP)       ;SET UP RETURN PC
005074  000002                          RTI                     ;CLEAR PSW AND CONTINUE
005076  000137  001702          1$:     JMP     QUETST
```

```
                                ;TYPQ4, ROUTINE, LOADS MSGD$ INFORMATION INTO TYPEQ

005102  004737  005350  TYPQ4,: JSR     PC,LDTYPQ       ;QUEUE PC+2 OF CALL
005106  013700  001122          MOV     MODADR,R0       ;LOAD MODULE ADDRESS IN R0
005112  010046                  MOV     R0,-(SP)        ;AND SAVE IN QUEUE
005114  004737  005350          JSR     PC,LDTYPQ
005120  016046  000060          MOV     AWAS(R0),-(SP)  ;QUEUE DATA ERROR COUNT
005124  004737  005350          JSR     PC,LDTYPQ
005130  016046  000056          MOV     ASB(R0),-(SP)   ;QUEUE SIZE OF TRANSFER
005134  004737  005350          JSR     PC,LDTYPQ
005140  005726                  POPSP                   ;REMOVE PS FROM STACK
005142  013706  001136          MOV     SPSAV,R6        ;GET MONITOR STACK POINTER
005146  000747                  BR      EXIT2,          ;EXIT
```



```
                                ;TYPQ3, ROUTINE QUEUES INFORMATION FOR DERRX CALLS

005150  011646          TYPQ3,: MOV     (SP),-(SP)
005152  004737  005350          JSR     PC,LDTYPQ       ;QUEUE PC+2 OF CALL
005156  013700  001122          MOV     MODADR,R0       ;LOAD MODULE ADDRESS IN R0
005162  010046                  MOV     R0,-(SP)        ;AND SAVE IN QUEUE
005164  004737  005350          JSR     PC,LDTYPQ
005170  017616  000000          MOV     @(SP),(SP)      ;QUEUE VIRTUAL ADDRESS OF CHECKDATA CALL
005174  004737  005350          JSR     PC,LDTYPQ
005200  005726                  POPSP
005202  062700  000062          ADD     #AWAS+2,R0      ;GET ADDRESS OF INFORMATION
005206  012701  000005          MOV     #5,R1
005212  000401                  BR      2$
005214  005010          1$:     CLR     (R0)            ;CLEAR DATA (EXCEPT LAST)
005216  014046          2$:     MOV     -(R0),-(SP)     ;GET DATA
005220  004737  005350          JSR     PC,LDTYPQ       ;QUEUE DATA
005224  005301                  DEC     R1
005226  001372                  BNE     1$
005230  013700  001122          MOV     MODADR,R0
005234  010660  000046          MOV     R6,SVR6(R0)
005240  013706  001136          MOV     SPSAV,R6
005244  000137  005066          JMP     EXIT2,
```

```
 1074                                   ;TYPQ2, ROUTINE SERVICES ERROR, DATA ERROR, AND ERRN CALLS.
 1075
 1076  005250  011646                   TYPQ2.: MOV     (SP),-(SP)      ;SAVE PC+2 OF CALL AGAIN.
 1077  005252  004737  005350                   JSR     PC,LDTYPQ       ;QUEUE UP PC+2 OF CALL.
 1078  005256  010046                           MOV     R0,-(SP)                ;SAVE R0.
 1079  005260  017600  000002                   MOV     @2(SP),R0       ;GET MODULE ADDR.
 1080  005264  062700  000050           TYPQX:  ADD     #CSRA,R0        ;
 1081  005270  012746  000005                   MOV     #5,-(SP)
 1082  005274  012046                           MOV     (R0)+,-(SP)
 1083  005276  000402                           BR      2$
 1084  005300  011046                   1$:     MOV     (R0),-(SP)
 1085  005302  005020                           CLR     (R0)+
 1086  005304  004737  005350           2$:     JSR     PC,LDTYPQ
 1087  005310  005316                           DEC     (SP)
 1088  005312  001372                           BNE     1$
 1089  005314  005726                           POPSP
 1090  005316  012600                           MOV     (SP)+,R0                ;RESTORE R0.
 1091  005320  000137  005030                   JMP     EXIT.
```



```
 1092                                   ;TYPQ, ROUTINE, SERVICES END, ENDPAS, AND MSGN CALLS.
 1093
 1094  005324  004737  005350           TYPQ.:  JSR     PC,LDTYPQ       ;QUEUE UP CALL.
 1095  005330  005726                           POPSP
 1096  005332  000137  005032                   JMP     EXIT1.
```

```
 1097                                   ;TYPQ1. ROUTINE. SERVICES MSG CALL.
 1098
 1099  005336  004737  005350          TYPQ1.: JSR     PC,LDTYPQ       ;QUEUE UP CALL.
 1100  005342  005726                          POPSP
 1101  005344  000137  005066                  JMP     EXIT2.
```



```
 1102                                   ;ROUTINE TO LOAD TYPE QUEUE. ENTERED VIA JSR PC.
 1103  005350  005046                  LDTYPQ: CLR     -(SP)           ;CLEAR PSW ON STACK
 1104  005352  012746  005360                  MOV     #3$,-(SP)       ;SETUP RETURN PC
 1105  005356  000002                          RTI                     ;CLEAR PSW AND CONTINUE
 1106  005360  105737  001007          3$:     TSTB    TYPQUE          ;REQUEST COUNT 0?
 1107  005364  001414                          BEQ     1$              ;BR IF YES.
 1108  005366  023737  001100  001102          CMP     TYPQ1,TYPQ2     ;NO. TYPQ1 AND TYPQ2 SAME?
 1109  005374  001010                          BNE     1$              ;BR IF NOT.
 1110  005376  012737  017116  001100          MOV     #TYPEQ,TYPQ1    ;RESET THE TYPE QUEUE
 1111  005404  104406  014157                  MSG$,QOVRFL     ;YES. QUEUE OFLO. CRASH SYSTEM
 1112  005410  104406  014620                  MSG$,HLT        ;'HALT'
 1113  005414  000000                          HALT
 1114  005416  023727  001100  017572  1$:     CMP     TYPQ1,#TYPLIM   ;REACHED HIGH LIMIT?
 1115  005424  001003                          BNE     2$              ;BR IF NOT.
 1116  005426  012737  017116  001100          MOV     #TYPEQ,TYPQ1    ;RESET TYPQ1.
 1117  005434  016677  000002  173436  2$:     MOV     2(SP),@TYPQ1    ;STORE ARGUMENT IN QUEUE
 1118  005442  105237  001007                  INCB    TYPQUE          ;UPDATE REQUEST COUNTS.
 1119  005446  062737  000002  001100          ADD     #2,TYPQ1        ;UPDATE TYPQ1.
 1120  005454  012616                          MOV     (SP)+,(SP)      ;
 1121  005456  000207                          RTS     PC              ;EXIT.
```

```
                                        ;ROUTINE TO LOAD BREAK$ CALL
                                        ;
005460  112766  177777  000003  LDBRK,: MOVB    #-1,3(SP)       ;INDICATE BREAK$ CALL
005466  000402                         BR      LDIOQ
                                        ;
                                        ;ROUTINE TO LOAD QUE$ CALL.
                                        ;
005470  005066  000002          QUE.:   CLR     2(SP)           ;INDICATE QUE CALL.
                                        ;
                                        ;LDIOQ ROUTINE. MUST NOT DROP BELOW LEVEL OF INTERRUPTING DEVICE.
                                        ;
005474  012746  000340          LDIOQ:  MOV     #PRTY7,-(SP)    ;PUT NEW PSW ON STACK
005500  012746  005506                  MOV     #1$,-(SP)       ;SETUP RETURN PC
005504  000002                         RTI                     ;LOAD NEW PSW AND CONTINUE
005506  000403                  1$:    BR      .+10
005510  112766  000001  000002 PIRQ.,: MOVB    #1,2(SP)
005516  105737  001006                  TSTB    IOQUE           ;REQUEST COUNT 0?
005522  001414                         BEQ     1$              ;BR IF YES.
005524  023737  001074  001076          CMP     IOQ1,IOQ2       ;IOQ1 AND IOQ2 SAME?
005532  001010                         BNE     1$              ;BR IF NOT.
005534  012737  016606  001074          MOV     #IOQ,IOQ1       ;RESET IOQ1
005542  104406  014157                  MSG$,QOVRFL     ;QUE OFLO. CRASH SYSTEM
005546  104406  014620                  MSG$,HLT        ;'HALT'
005552  000000                         HALT
005554  023727  001074  017116  1$:    CMP     IOQ1,#IOQLIM    ;REACHED HIGH LIMIT?
005562  001003                         BNE     2$              ;BR IF NOT.
005564  012737  016606  001074          MOV     #IOQ,IOQ1       ;RESET IOQ1.
005572  012677  173276          2$:    MOV     (SP)+,@IOQ1     ;STORE PC+2 OF PENDING CALL.
005576  105237  001006                  INCB    IOQUE           ;UPDATE REQUEST COUNTS.
005602  062737  000002  001074          ADD     #2,IOQ1         ;UPDATE IOQ1.
005610  005716                         TST     (SP)            ;QUE CALL?
005612  001003                         BNE     3$              ;NO- BRANCH
005614  005726                         TST     (SP)+           ;YES, POP STACK
005616  000137  005066                  JMP     EXIT2.          ;EXIT
005622  126627  000001  177777  3$:    CMPB    1(SP),#-1       ;BREAK CALL?
005630  001003                         BNE     4$              ;NO- BRANCH
005632  005726                         TST     (SP)+           ;YES- POP STACK
005634  000137  005032                  JMP     EXIT1.
005640  005726                  4$:    TST     (SP)+           ;PIRQ- EXIT VIA RTI
005642  000002                         RTI
```



```
                                        ;KEYBOARD INTERRUPT SERVICE
                                        ;
005644  005077  173276          KBSRVC: CLR     @TKS            ;CLEAR INTERRUPT ENABLE.
005650  017746  173274                  MOV     @TKB,-(SP)      ;GET CHAR.
005654  042716  000200                  BIC     #200,(SP)       ;REMOVE EXTRA BITS.
005660  122716  000003                  CMPB    #3,(SP)         ;IS IT CTRL C?
005664  001524                         BEQ     CTRLCA          ;BR IF YES.
005666  005726                         POPSP                   ;REMOVE CHAR FROM STACK.
005670  000004  005676  000000          PIRQ$,1$,0              ;Q DEVICE SERVICE.
                                        ;
                                        ;KEYBOARD SERVICE.
                                        ;
005676  023727  001036  016346  1$:    CMP     KBPTR,#KBUF+KBUFL ;BUFFER EXCEEDED?
005704  001511                         BEQ     12$             ;BR IF YES.
005706  017746  173236                  MOV     @TKB,-(SP)      ;GET CHAR.
005712  042716  000200                  BIC     #200,(SP)       ;CLEAR PARITY BIT.
005716  121627  000177                  CMPB    (SP),#177       ;RUBOUT?
005722  001443                         BEQ     5$              ;BR IF YES.
005724  121627  000040                  CMPB    (SP),#40                ;SPACE? (IGNORE LEADING ONES).
005730  001460                         BEQ     8$              ;BR IF YES.
005732  121627  000015                  CMPB    (SP),#15                ;CR? (TIME TO QUIT?).
005736  001462                         BEQ     9$              ;BR IF YES.
005740  121627  000012                  CMPB    (SP),#12                ;LF? (TREAT LIKE CR).
005744  001457                         BEQ     9$              ;BR IF YES.
005746  121627  000060                  CMPB    (SP),#60                ;CHAR LESS THAN 60?
005752  103425                         BLO     10$             ;BR IF YES, IGNORE IT.
005754  121627  000141                  CMPB    (SP),#141       ;CHAR LESS THAN LOWER CASE A?
005760  103405                         BLO     2$              ;BR IF YES.
005762  121627  000172                  CMPB    (SP),#172       ;CHAR HIGHER THAN LOWER CASE Z?
005766  101017                         BHI     10$             ;BR IF YES.
005770  162716  000040                  SUB     #40,(SP)                ;LOWER CASE CHAR, MAKE IT UPPER CASE.
005774  111677  173036          2$:    MOVB    (SP),@KBPTR     ;STORE CHAR.
006000  005237  001036                  INC     KBPTR           ;UPDATE BUFFER POINTER.
006004  112637  001240          3$:    MOVB    (SP)+,TTYBYT    ;CHAR TO ECHO BUFFER.
006010  104406  001240                  MSG$,TTYBYT             ;ECHO THE CHARACTER.
006014  052777  000100  173124  4$:    BIS     #IE,@TKS        ;REENABLE KYBD INTERRUPTS.
006022  000137  001702                  JMP     QUETST
006026  005726                  10$:    POPSP
006030  000771                         BR      4$
                                        ;
                                        ;SERVICE DELETE/RUBOUT CODE.
                                        ;
006032  022737  016306  001036  5$:    CMP     #KBUF,KBPTR     ;BUFFER EMPTY?
006040  001405                         BEQ     6$              ;BR IF YES.
006042  005337  001036                  DEC     KBPTR           ;POINT TO PREVIOUS CHAR.
006046  117716  172764                  MOVB    @KBPTR,(SP)     ;GET PREVIOUS CHAR.
006052  000754                         BR      3$              ;GO ECHO IT.
006054  005726                  6$:    POPSP
006056  104406  014155                  MSG$,ACRLF              ;OUTPUT CRLF.
006062  012737  016306  001036  7$:    MOV     #KBUF,KBPTR     ;RESET KBUF POINTER.
006070  000751                         BR      4$
                                        ;
                                        ;SERVICE SPACE.
                                        ;
006072  022737  016306  001036  8$:    CMP     #KBUF,KBPTR     ;LEADING SPACE?
006100  001741                         BEQ     3$              ;BR IF YES (IGNORE IT).
```

```
1218  006102  000734                          BR      2$              ;STORE IT.
1219
1220                                  ;SERVICE CR AND LF.
1221
1222  006104  112677  172726          9$:     MOVB    (SP)+,@KBPTR    ;STORE CHAR.
1223  006110  012737  016306  001036          MOV     #KBUF,KBPTR     ;POINT TO START OF KBUF.
1224  006116  104406  014155                  MSG$,ACRLF              ;OUTPUT CRLF
1225  006122  104401  007230  000000          QUE$,DECODE,0           ;QUE TO SERVICE COMMAND.
1226
1227                                  ;SERVICE KBUF OVERFLOW.
1228
1229  006130  104406  014416          12$:    MSG$,KBOFLO             ;TYPE KBUF OFLO.
1230  006134  000752                          BR      7$
```



```
1231                                  ;SERVICE CTRL C, ENDS RUN MODE ALSO,
1232                                  ;DROPS TO BANKS 0 AND 1 BEFORE RETURNING (IF KT11 IN USE)
1233                                  ;STACK POINTER SHOULD BE SETUP AHEAD OF CALL TO CTRLX TO PRESERVE
1234                                  ;NEEDED INFORMATION PRIOR TO REMAPPING TO 0, OR CAUSE REMAPPING TO
1235                                  ;CURRENT AREA AFTER
1236
1237  006136  012706  016546          CTRLCA: MOV     #SPBOT,SP       ;REINITIALIZE STACK POINTER
1238  006142  004737  006230          CTRLCB: JSR     PC,CTRLX        ;CLEAR QUEUES, TYPE ^C
1239  006146  005037  001224                  CLR     OFFSTD          ;GET PROGRAM BACK TO BANK 0
1240  006152  004737  014060                  JSR     PC,MVCODE
1241  006156  105737  001257                  TSTB    RMODE           ;IN RUN MODE?
1242  006162  001002                          BNE     1$              ;BR IF YES,
1243  006164  000137  007374                  JMP     COMCO3          ;BACK TO KYBD ROUTINE,
1244  006170  105037  001257          1$:     CLRB    RMODE           ;CLEAR RUN MODE INDICATOR
1245  006174  105737  001260                  TSTB    CHN             ;IN CHAIN MODE?
1246  006200  100411                          BMI     CTRLCD          ;BR IF YES, BYPASS SUMMARY AND RETURN TO
1247                                                                  ;CHAIN MONITOR
1248  006202  052777  000100  172736          BIS     #IE,@TKS        ;REENABLE KEYBOARD INTERRUPTS
1249  006210  104406  014136                  MSG$,SUMARY             ;TYPE RUN END SUMMARY TITLE,
1250  006214  104406  014125          2$:     MSG$,CR                 ;CARRIAGE RETURN
1251  006220  004737  010274                  JSR     PC,DIRA         ;TYPE RUN SUMMARY,
1252  006224  000137  012722          CTRLCD: JMP     CHNOUT          ;EXIT, OR RETURN TO KYBD RTN,
1253
```

```
1254  006230  012746  000340          CTRLX:  MOV     #PRTY7,-(SP)    ;SETUP NEW PSW ON STACK
1255  006234  012746  006242                  MOV     #48,-(SP)       ;PUT RETURN PC ON STACK
1256  006240  000002                          RTI                     ;LOAD NEW PSW AND CONTINUE
1257  006242  012737  000144  001374  48$:    MOV     #100.,CSTART    ;CLEAR QUEUES AND DELAY TOO.
1258
1259  006250  105737  001257                  TSTB    RMODE           ;IF NOT IN RMODE, SHORTEN WAIT
1260  006254  001003                          BNE     1$
1261  006256  012737  000010  001374          MOV     #10,CSTART
1262  006264  004737  012272          1$:     JSR     PC,CLRQUS       ;CLEAR QUEUES.
1263  006270  005337  001374                  DEC     CSTART          ;DONE?
1264  006274  001373                          BNE     1$              ;BR IF NOT.
1265  006276  011604                          MOV     (SP),R4         ;SAVE RETURN PC
1266  006300  000005                          RESET
1267  006302  010416                          MOV     R4,(SP)         ;RESTORE RETURN PC IN CURRENT BANK
1268  006304  004737  012272                  JSR     PC,CLRQUS       ;CLEAR QUEUES IN BANK 0
1269  006310  005046                  38$:    CLR     -(SP)           ;RETYPE NEW PSW ON STACK
1270  006312  012746  006320                  MOV     #56,-(SP)       ;PUT RETURN PC ON STACK
1271  006316  000002                          RTI                     ;LOAD NEW PSW AND CONTINUE
1272  006320  105237  001013          56$:    INCB    BRAKE           ;SET BRAKE TO PREVENT BACKGROUND MODULE FROM BEING START
1273  006324  013737  001266  001264          MOV     ROTNUM,ROTCNT   ;RESET THE ROTATION COUNTER
1274  006332  104406  014153                  MSG$,CTRLC              ;OUTPUT ^C
1275  006336  000207                          RTS     PC              ;EXIT.
```



```
1276                                    ;SUBROUTINE TO TYPE MULTIPLE ASCII STRINGS. NOTE THAT THE PREVIOUS
1277                                    ;VALUE OF R5 IS NOT SAVED.
1278
1279  006340  005726                  TYPEN:  TST     (SP)+           ;REMOVE OLD R5 FROM STACK
1280  006342  012546                  TYPE1:  MOV     (5)+,-(SP)      ;GET ASCII ADDRESS.
1281  006344  001403                          BEQ     1$              ;BR IF 0 TERMINATOR.
1282  006346  004737  006356                  JSR     PC,TYPE         ;TYPE IT.
1283  006352  000773                          BR      TYPE1           ;GET MORE.
1284  006354  000205                  1$:     RTS     R5              ;RETURN.
```

```
                                        ;TYPE SUBROUTINE,
                                        ;TYPES ASCIZ STRING
                                        ;NOTE THAT IF RMODE IS SET BUT MODULES AREN'T YET BEING STARTED, BRAKE
                                        ;MUST BE SET TO PREVENT STARTNG MODULES WHILE TYPING

006356  012737  006426  001410  TYPE:  MOV     #TTSRV1,TTYLNK
006364  012637  001106                 MOV     (SP)+,TYPRET    ;SAVE RETURN ADDRESS
006370  012637  001104                 MOV     (SP)+,TYPADR    ;SAVE MESSAGE ADDRESS
006374  010137  001124                 MOV     R1,TYPR1        ;SAVE REGISTERS 1-5
006400  012701  001126                 MOV     #TYPR2,R1
006404  010221                        MOV     R2,(R1)+
006406  010321                        MOV     R3,(R1)+
006410  010421                        MOV     R4,(R1)+
006412  010521                        MOV     R5,(R1)+
006414  012777  000100  172530        MOV     #100,@TPS       ;SET IE

006422  000137  001702                 JMP     QUETST
006426  117746  172452        TTSRV1: MOVB    @TYPADR,-(SP)   ;GET CHAR.
006432  001015                        BNE     TYPEB           ;BR IF NOT TERMINATOR.
006434  005726                        TST     (SP)+
006436  012701  001126                 MOV     #TYPR2,R1       ;RESTORE REGISTERS 1-5
006442  012102                        MOV     (R1)+,R2
006444  012103                        MOV     (R1)+,R3
006446  012104                        MOV     (R1)+,R4
006450  012105                        MOV     (R1)+,R5
006452  013701  001124                 MOV     TYPR1,R1
006456  005077  172470                 CLR     @TPS            ;CLEAR IE
006462  000177  172420                 JMP     @TYPRET

006466  062737  000001  001104  TYPEB: ADD     #1,TYPADR
006474  122716  000045                 CMPB    #45,(SP)                ;IS IT %?
006500  001037                        BNE     TYPEL           ;NO- BRANCH
006502  112716  000015        TYPEC:  MOVB    #15,(SP)                ;OUTPUT CR.
006506  012737  006516  001410        MOV     #1$,TTYLNK
006514  000431                        BR      TYPEL
006516  112746  000012        1$:     MOVB    #12,-(SP)       ;OUTPUT LF.
```



```

006522  012737  006532  001410        MOV     #2$,TTYLNK
006530  000423                        BR      TYPEL
006532  113737  001254  001020  2$:   MOVB    FILCNT,FILCTR   ;GET FILL COUNT.
006540  001002                        BNE     TYPEE           ;BR IF NOT 0.
006542  105237  001020                 INCB    FILCTR          ;OOPS, MAKE IT A 1.
006546  113746  001255        TYPEE:  MOVB    FILLER,-(SP)    ;OUTPUT FILLER.
006552  012737  006562  001410  TYPED: MOV     #3$,TTYLNK
006560  000407                        BR      TYPEL
006562  105337  001020          3$:   DECB    FILCTR          ;DECREMENT FILL COUNTER.
006566  001367                        BNE     TYPEE           ;BR IF NOT 0.
006570  012737  006426  001410        MOV     #TTSRV1,TTYLNK
006576  000713                        BR      TTSRV1
006600  012677  172350        TYPEL:  MOV     (SP)+,@TPB
006604  012737  010000  001110        MOV     #10000,TYPCTR
006612  000137  001702                 JMP     QUETST
```

```
                                        ;ROUTINE TO SERVICE CKDATA CALL (CDATA$)
                                
006616  112737  177777  001315  DATCK.: MOVB    #-1,DATFLG      ;SET FLAG FOR DATACK CALL
006624  000402                          BR      CKCONT          ;CONTINUE
006626  105037  001315          CDATA.: CLRB    DATFLG          ;SET FLAG FOR CKDATA CALL
                                
006632  005037  001142          CKCONT: CLR     CDFLG
006636  005037  001114                  CLR     DERRCT          ;INITIALIZE DATA ERROR COUNT
006642  004437  012244                  JSR     R4,$AV04        ;TEMPORARILY SAVE REGS
006646  016604  000012                  MOV     12(SP),R4       ;GET PC+2 OF CALL
006652  010437  001400          CDATA1: MOV     R4,CDERR1       ;SETUP PC OF CALL
006656  162737  000002  001400          SUB     #2,CDERR1
006664  012401                          MOV     (R4)+,R1        ;GET BEGIN ADDRESS OF MODULE
006666  012403                          MOV     (R4)+,R3        ;GET TABLE ADDRESS OF RBUFPA FROM MODULE
006670  010361  000052                  MOV     R3,SBADR(R1)    ;SAVE IT
006674  062701  000032                  ADD     #SVR0,R1        ;SAVE MODULE'S REGISTERS
006700  012621                          MOV     (SP)+,(R1)+
006702  012621                          MOV     (SP)+,(R1)+
006704  012621                          MOV     (SP)+,(R1)+
006706  012621                          MOV     (SP)+,(R1)+
006710  012621                          MOV     (SP)+,(R1)+
006712  010511                          MOV     R5,(R1)
006714  022626                          CMP     (SP)+,(SP)+
006716  016401  177774                  MOV     -4(R4),R1       ;RESTORE MODULE ADDR
006722  012300                          MOV     (3)+,R0         ;GET READ BUFFER PA
006724  005723                          TST     (3)+            ;SKIP EA BITS
006726  012305                          MOV     (3)+,R5         ;BUFFER WORD COUNT
006730  016102  000074                  MOV     WBUFPA(R1),R2   ;GET WRITE BUFFER PA
006734  005003                          CLR     R3              ;CLEAR OUT REG, 3
006736  005737  001142                  TST     CDFLG           ;ENTERED FROM DATA ERROR RETURN?
006742  001413                          BEQ     30$             ;NO- BRANCH
006744  012603                          MOV     (SP)+,R3        ;YES- GET COUNT
006746  005203                          INC     R3
006750  006303                          ASL     R3              ;CREATE OFFSET
006752  060300                          ADD     R3,R0           ;GET ADDRESS OF CURRENT READ BUFFER WORD
006754  103003                          BCC     .+10
006756  060302                          ADD     R3,R2           ;GET ADDRESS OF CURRENT WRITE BUFFER WORD
006760  103003                          BCC     .+10
006762  006203                          ASR     R3              ;RESTORE COUNT
006764  160305                          SUB     R3,R5           ;CALCULATE WORDS LEFT
006766  012637  001114                  MOV     (SP)+,@DERRCT   ;RESTORE DATA ERROR COUNT
006772                          30$:
006772  021012                  1$:     CMP     @R0,@R2         ;DATA OK?
006774  001427                          BEQ     3$              ;YES, BRANCH
006776  005237  001114                  INC     DERRCT          ;NO, COUNT DATA ERROR
007002  023727  001114  000003          CMP     DERRCT,#3       ;MORE THAN 3 ERRORS?
007010  003404                          BLE     2$              ;NO- REPORT ERROR
007012  032777  002000  172136          BIT     #BIT10,@$SR     ;YES- REPORT ALL?
007020  001415                          BEQ     3$              ;NO- BRANCH
007022  011061  000060          2$:     MOV     @R0,AWAS(R1)    ;SAVE ACTUAL DATA
007026  011261  000056                  MOV     @R2,ASB(R1)     ;SAVE EXPECTED DATA
007032  010361  000054                  MOV     R3,WASADR(R1)   ;SAVE COUNT
007036  013746  001114                  MOV     DERRCT,-(SP)    ;SAVE DATA ERROR COUNT
007042  010346                          MOV     R3,-(SP)        ;SAVE WORD COUNT ON STACK
007044  013746  001400                  MOV     CDERR1,-(SP)    ;SAVE VA OF CKDATA CALL
007050  000137  001376                  JMP     CDERR           ;GO TO ERROR CALL
```



```
007054  005203                  3$:     INC     R3              ;COUNT WORDS CHECKED
007056  005720                          TST     (R0)+
007060  005722                          TST     (R2)+
007062  005305                  5$:     DEC     R5
007064  001342                          BNE     1$
007066  005737  001114                  TST     DERRCT          ;ANY DATA ERRORS?
007072  001003                          BNE     6$              ;YES, BRANCH
007074  012437  001054                  MOV     (R4)+,@STADR    ;GET THE RETURN ADDRESS
007100  000423                          BR      7$              ;GO TO COMMON RETURN CODE
007102  005261  000030          6$:     INC     ERCNT(R1)       ;DATA ERRORS OCCURRED- UP MODULE ERROR
                                                                ;COUNT BY 1
007106  001002                          BNE     .+6
007110  005161  000030                  COM     ERCNT(R1)       ;SET TO -1 IF 0
007114  013761  001114  000060          MOV     DERRCT,AWAS(R1) ;STORE INFORMATION IN MODULE
007122  010361  000056                  MOV     R3,ASB(R1)      ;STORE SIZE IN ASB
007126  010446                          MOV     R4,-(SP)        ;SAVE POINTER TO ERROR RETURN ADDRESS
007130  010661  000046                  MOV     SP,SVR6(R1)     ;SAVE STACK POINTER
007134  104422                          MSGD$                   ;REPORT TOTALS, HALT MODULE IF NEEDED
007136  012604                          MOV     (SP)+,R4        ;GET ERROR RETURN POINTER
007140  013701  001122                  MOV     MODADR,R1       ;GET MODULE START ADDRESS
007144  012437  001054                  MOV     (R4)+,@STADR    ;GET THE RETURN ADDRESS
007150  105737  001315          7$:     TSTB    DATFLG          ;IS THIS A DATACK CALL ?
007154  100002                          BPL     8$              ;NO, CONTINUE
007156  013734  001114                  MOV     DERRCT,@(R4)+   ;YES, LOAD THE NUMBER OF DATA ERRORS
007162  116137  000020  001066  8$:     MOVB    STAT(R1),@RSTAT ;SET UP FOR MODULE RETURN
007170  010100                          MOV     R1,R0
007172  062700  000046                  ADD     #SVR6,R0
007176  000137  002570                  JMP     IDQSVD
                                
007202  005746                  CDRET:  TST     -(SP)           ;PUSH STACK
007204  004437  012244                  JSR     R4,$AV04
007210  016604  000014                  MOV     14(SP),R4       ;GET VA OF CKDATA CALL
007214  062704  000002                  ADD     #2,R4
007220  005237  001142                  INC     CDFLG
007224  000137  006652                  JMP     CDATA1
```

```
                        ;COMMAND DECODER. CHECKS SYNTAX AGAINST COMMAND TABLE.
                        ;ALL COMMANDS ARE CHECKED AGAINST THE QUOTES. NO ABBREVIATIONS ALLOWED

007230  105737  001010 DECODE: TSTB    SPCFLG          ;SPECIAL FLAG SET?
007234  001404         1$:     BEQ     1$              ;BR IF NOT.
007236  005037  001010         CLR     SPCFLG          ;YES. CLEAR IT.
007242  000177  171610         JMP     @SPRETRN        ;GO VIA SRETRN.
007246  105737  001257         TSTB    RMODE           ;IN RUN MODE?
007252  001403                 BEQ     2$              ;NO, CONTINUE
007254  104406  014304         MSG$,INVCD1             ;'INVALID CMD. ONLY CNTL C IS OK'
007260  000441                 BR      COMCON          ;CONTINUE
007262  004437  012244 2$:     JSR     R4,SAV04        ;SAVE REGS 0-4.
007266  013700  001036         MOV     KBPTR,R0        ;POINTER TO R0.
007272  012701  015232 GTOK:   MOV     #COMTAB,R1      ;DEVICE DECODING COMES HERE
007276  010046         GTOKX:  MOV     R0,-(SP)        ;SAVE STRING POINTER
007300  122021         GTOK1:  CMPB    (R0)+,(R1)+     ;MATCH THE CHARACTER?
007302  001407                 BEQ     GTOK2           ;YEP, SEE IF THE REST MATCH
007304  105721                 TSTB    (R1)+           ;NO, GO CHECK AGAINST NEXT
007306  001376                 BNE     .-2             ;BUT FIND WHERE NEXT ENTRY IS
007310  111102                 MOVB    (R1),R2         ;DISPLACEMENT TO NEXT ENTRY
007312  001422                 BEQ     COMCO1          ;NO SUCH THING.
007314  060201         MORE:   ADD     R2,R1           ;POINT TO NEXT ENTRY
007316  011600                 MOV     (SP),R0         ;POINT TO BEGINNING OF INPUT
007320  000767                 BR      GTOK1           ;KEEP LOOKING
007322  105711         GTOK2:  TSTB    (R1)            ;LAST CHR IN ENTRY?
007324  001365                 BNE     GTOK1           ;NO,CHECK THE OTHER CHR'S
007326  122126                 CMPB    (R1)+,(SP)+     ;RESET SP AND R1
007330  111102                 MOVB    (R1),R2         ;GET DISPATCH ADDRESS
007332  060201                 ADD     R2,R1           ;R1 POINTS TO NEXT ENTRY
007334  014137  001372         MOV     -(R1),CADDR
007340  005037  001374         CLR     CSTART          ;SET MODULE ADDR IN CSTART TO 0.
007344  010037  001036         MOV     R0,KBPTR        ;SAVE CURRENT BUFFER POINTER.
007350  004737  012256         JSR     PC,RST04        ;RESTORE REGS 0-4.
007354  000137  001362         JMP     COMQUE          ;GO TO COMMON QUE CALL.
```



```
                        ;COMMON RETURN FROM COMMAND EXECUTION AND RUN ROUTINE (INCLUDING
                        ;RELOCATION AND INTERNAL RESTART)

007360  104406  014272 COMCO1: MSG$,INVCMD              ;TYPE INVALID COMMAND.
007364         COMCON:
007364  005037  001010 COMCO2: CLR     SPCFLG          ;CLEAR SPECIAL FLAG AND
                                                        ;DIR COMMAND INDICATOR.
007370  105037  001272         CLRB    FILLID          ;CLEAR FILL COMMAND INDICATOR.
007374  104406  014122 COMCO3: MSG$,DOT                ;TYPE DOT.
007400  000137  001650 COMCO4: JMP     INPUT           ;GO GET MORE INPUT.
```

```
                                ;SPECIAL INPUT ROUTINE.
                                ;CALLED BY MODIFY ROUTINE, WHEN WAITING FOR NEW VALUE TO BE INPUT.

007404  012637  001056         SINPUT: MOV     (SP)+,SRETRN    ;SAVE RETURN ADDR.
007410  105237  001010                 INCB    SPCFLG          ;SET SPECIAL FLAG.
007414  000771                         BR      COMCO4
```



```
                                ;RUN ROUTINE, STARTS EXERCISER EXECUTION.
                                ;NOTE THAT "RUN" IS ENTERED ONLY FROM BANK 0, BUT "RUNRES" IS USED WHEN
                                ;THE PROGRAM IS NOT IN BANK 0.

007416  000402                         BR      .+6
007420  105037  001275         RUN:    CLRB    LOCK            ;CLEAR LOCK TO ALLOW CODE RELOCATION
007424  005037  001224                 CLR     OFFSTD
007430  005037  001244                 CLR     SYSCNT          ;ZERO SYSTEM ERROR COUNTER
007434  005037  001246                 CLR     PWRCNT          ;ZERO POWER FAIL COUNTER
007440  012737  042761  010470  2$:    MOV     #42761,SLDSR    ;RESTORE THESE  INSTRUCTIONS
007446  012737  042761  010512         MOV     #42761,SLDSE    ;FOR A CHECK SUM MODULE
007454  105037  001274                 CLRB    PUNPFL          ;
007460  105037  001257                 CLRB    RMODE           ;
007464  012737  016234  001250         MOV     #MODQ-2,BKPTR
007472  004737  012354                 JSR     PC,CLRVEC       ;INITIALIZE TRAPCATCHER IN VECTOR AREA
007476  004737  012272         RUNRES: JSR     PC,CLRQUS       ;INCLUDES CLEARING BRAKE,MODCNT,IORKID
007502  112737  177777  001023  11$:    MOVB    #-1,WBFLG       ;INITIALIZE BUFFER ROTATION FLAG
007510  012702  016236                 MOV     #MODQ,R2        ;SETUP TO COUNT SELECTED MODULES, AND
                                                                ;IF NOT SYSERR OR POWER FAIL RESTART
                                                                ;CLEAR MODULES' PASCNT AND ERRCNT.
007514  012201                 1$:     MOV     (2)+,R1         ;MODULE ADDR TO R1.
007516  001462                         BEQ     RUNC            ;BR IF NO MORE.
007520  105737  001257                 TSTB    RMODE           ;INTERNAL RESTART?
007524  001012                         BNE     2$              ;BR IF YES.
007526  105737  001260                 TSTB    CHN             ;IN CHAIN MODE?
007532  100407                         BMI     2$              ;BR IF YES.
007534  005061  000026                 CLR     PSCNT(R1)       ;CLEAR MOD'S PASCNT.
007540  005061  000030                 CLR     ERCNT(R1)       ;CLEAR MOD'S ERROR COUNT.
007544  042761  020000  000020         BIC     #BIT13,STAT(R1) ;CLEAR STOPPED BIT.
007552  032761  040000  000020  2$:    BIT     #BIT14,STAT(R1) ;MODULE SELECTED?
007560  001755                         BEQ     1$              ;BR IF NOT.
007562  032761  020000  000020         BIT     #BIT13,STAT(R1) ;MODULE STOPPED?
007570  001351                         BNE     1$              ;BR IF YES.
007572  005737  001222                 TST     OFFSET          ;IF OFFSET 0, IGNORE BITS 10+11
007576  001414                         BEQ     3$
007600  032761  004000  000020         BIT     #BIT11,STAT(R1) ;NOT 0= IF BIT11 IS SET, DON'T COUNT IT
007606  001342                         BNE     1$
007610  032737  001777  001222         BIT     #1777,OFFSET    ;32K BOUNDARY?
007616  001404                         BEQ     3$              ;YES= IGNORE BIT 10
007620  032761  002000  000020         BIT     #BIT10,STAT(R1) ;NO= BIT 10 SET?
007626  001332                         BNE     1$              ;YES= DON'T RUN THIS MODULE
007630  105237  001015         3$:     INCB    MODCNT          ;NO, UP COUNT OF RUNABLE MODULES.
007634  005761  000020                 TST     STAT(R1)                ;BKMOD?
007640  100402                         BMI     .+6             ;NO= BRANCH
007642  105237  001025                 INCB    BKCNT           ;YES= UP COUNT
007646  032761  010000  000020         BIT     #BIT12,STAT(R1) ;EXTENDED MODULE?
007654  001402                         BEQ     .+6             ;NO= BRANCH
007656  105237  001026                 INCB    XCNT            ;YES= COUNT IT
007662  000714                         BR      1$              ;GO CHECK NEXT MODULE.
007664  105737  001015         RUNC:   TSTB    MODCNT          ;ANY RUNABLE MODULES?
007670  001005                         BNE     5$              ;YES, BRANCH
007672  005737  001222                 TST     OFFSET          ;NO, OFFSET 0?
007676  001501                         BEQ     RUND            ;YES= COMMAND INVALID
007700  000137  004604                 JMP     ENDSVB          ;NO= THE ONLY MODS RUNNABLE MUST BE THOSE
                                                                ;WITH BITS 10 OR 11 SET= RELOCATE
007704  113737  001015  001016  5$:    MOVB    MODCNT,MODCTR   ;LOAD COUNT OF RUNNABLE MODULES
```

```
                                                        ;INTO STARTUP COUNTER
007712  012737  016234  001040          MOV     #MODQ-2,MODPTR  ;MODULE TABLE ADDR TO MODPTR,
007720  012737  040000  001060          MOV     #40000,IOBKID   ;START WITH SPECIAL BACKGROUND MODULES,
007726  105237  001320                  INCB    NSTOP           ;SET THE NON-STOP FLAG
007732  105737  001261                  TSTB    ROTI
007736  001447                          BEQ     2$
007740  113737  001026  001031          MOVB    XCNT,ROTCT      ;SETUP ROTATION TIMEOUT COUNTER- 16 TIMES NUMBER
                                                        ;OF EXTENDED MODULES
007746  106337  001031                  ASLB    ROTCT
007752  106337  001031                  ASLB    ROTCT
007756  106337  001031                  ASLB    ROTCT
007762  106337  001031                  ASLB    ROTCT
007766  123727  001026  000020          CMPB    XCNT,#20        ;PREVENT LOSS OF COUNT IF OVERFLOW
007774  103403                          BLO     1$
007776  112737  000377  001031          MOVB    #377,ROTCT
010004  113737  001031  001302  1$:     MOVB    ROTCT,$GWBFC    ;INITIALIZE WRITE BUFFER ROTATION COUNTER
010012  113737  001015  001033          MOVB    MODCNT,PASREL   ;SET ALTERNATE RELOCATION COUNT TO 8
010020  106337  001033                  ASLB    PASREL          ;TIMES MODCNT
010024  106337  001033                  ASLB    PASREL
010030  106337  001033                  ASLB    PASREL
010034  123727  001015  000040          CMPB    MODCNT,#40
010042  103403                          BLO     3$
010044  112737  000377  001033          MOVB    #377,PASREL
010052  004737  010126          3$:     JSR     PC,SETWBF
010056  105737  001257          2$:     TSTB    RMODE
010062  001005                          BNE     4$
010064  104406  014122                  MSGS,DOT
010070  112737  000001  001257          MOVB    #1,RMODE        ;SET RMODE IN CURRENT BANK
010076  000137  001650          4$:     JMP INPUT
010102  005037  001224          RUND:   CLR     OFFSTD
010106  004737  014060                  JSR     PC,MVCODE
010112  105037  001257                  CLRB    RMODE
010116  104406  014346                  MSGS,INVCD4             ;'NO MODULES SELECTED'
010122  000137  007364                  JMP     COMCON
```



```
                        ;SETUP WRITE BUFFER INFORMATION

010126  005037  000076  SETWBF: CLR     WBUFEA          ;ZERO EA BITS
010132  013737  001200  001212          MOV     LOCORE,WBUF     ;LOAD STARTING ADR OF WRITE BUFFER
010140  013737  001204  001216          MOV     LCOR12,WBUF12
010146  063737  001222  001216          ADD     OFFSET,WBUF12
010154  023737  001216  001206          CMP     WBUF12,HCOR12   ;IS WRITE BUFFER AT THE TOP ?
010162  001001                          BNE     SETWB1          ;NO, IT'S OK, CONTINUE
010164  000207                          RTS     PC              ;RETURN

010166  013737  001206  001236  SETWB1: MOV     HCOR12,MXWBF    ;CALCULATE MAX SIZE OF WRITE BUFFER
010174  105737  001260                  TSTB    CHN             ;IN CHAIN MODE ?
010200  100003                          BPL     1$              ;NO, DON'T ADD ANY MORE WRITE BUFFER
010202  062737  000040  001236          ADD     #40,MXWBF       ;YES, ADD 1K TO MAX, WRITE BUFFER SIZE
010210  163737  001216  001236  1$:     SUB     WBUF12,MXWBF
010216  006337  001236                  ASL     MXWBF
010222  006337  001236                  ASL     MXWBF
010226  006337  001236                  ASL     MXWBF
010232  006337  001236                  ASL     MXWBF
010236  006337  001236                  ASL     MXWBF
010242  103003                          BCC     .+10
010244  012737  177777  001236          MOV     #177777,MXWBF
010252  000207                          RTS     PC
```

```
                                ;MAP ROUTINE. TYPES RESIDENT MODULES AND THEIR START ADDRESS.
                                ;
010254  105237  001011          MAP:    INCB    DIRIND          ;SET DIR INDICATOR.
010260  105037  015450                  CLRB    MDIRE           ;TERMINATE ASCII STRING EARLY.
010264  004737  010274                  JSR     PC,DIRA         ;TYPE MAP.
010270  000137  007364                  JMP     COMCON
                                ;
010274  012702  016236          DIRA:   MOV     #MODQ,R2        ;GET MODULE TABLE ADDR.
010300  012201                  1$:     MOV     (2)+,R1         ;GET MODULE ADDR.
010302  001446                          BEQ     5$              ;BR IF 0. ALL DONE.
010304  032761  040000  000020          BIT     #BIT14,STAT(R1) ;MODULE SELECTED?
010312  001003                          BNE     2$              ;BR IF YES.
010314  105737  001011                  TSTB    DIRIND          ;TYPING DIRECTORY?
010320  001767                          BEQ     1$              ;BR IF NOT. DONT TYPE UNSELECTED MODS.
010322  004537  011206          2$:     JSR     R5,FILLNM       ;FILL MOD NAME IN ASCII STRING.
010326  015415                          AMODNM+1                ;ADDR TO STUFF NAME IN.
010330  010146                          MOV     R1,-(SP)                ;MODULE ADDR TO STACK.
010332  004537  011512                  JSR     R5,OACNV        ;CONVERT MOD ADDR TO ASCII.
010336  015426                          APC
010340  016146  000020                  MOV     STAT(R1),-(SP)  ;CONVERT MODULE STATUS.
010344  004537  011512                  JSR     R5,OACNV
010350  015442                          AMDSTA
010352  016146  000026                  MOV     PSCNT(R1),-(SP) ;MOD'S PASS COUNT TO STACK.
010356  004537  011744                  JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
010362  015461                          APSCNT
010364  016146  000030                  MOV     ERCNT(R1),-(SP) ;MOD'S ERROR COUNT TO STACK.
010370  004537  011744                  JSR     R5,BDCNV        ;CONVERT IT TO DECIMAL ASCII.
010374  015500                          AERRS
010376  105737  001011                  TSTB    DIRIND          ;TYPING DIRECTORY?
010402  001003                          BNE     3$              ;BR IF YES.
010404  112737  000040  015450          MOVB    #40,MDIRE       ;NO. ALLOW TYPING FULL STRING.
010412  104406  015414          3$:     MSG$,AMODNM             ;TYPE ASCII LINE.
010416  000730                          BR      1$              ;DO IT AGAIN.
010420  000207                  5$:     RTS     PC              ;EXIT.
```



```
                                ;SELECT MODULE(S) ROUTINE.
                                ;
010422  012737  052761  010470  SEL:    MOV     #52761,SLDSB    ;MODIFY COMMON TO SELECT MODULE(S).
010430  012737  052761  010512          MOV     #52761,SLDSE    ;
010436  000406                          BR      SLDS            ;GO TO COMMON.
                                ;
                                ;DESELECT MODULE(S) ROUTINE.
                                ;
010440  012737  042761  010470  DES:    MOV     #42761,SLDSB    ;MODIFY COMMON TO DESELECT MODULE(S).
010446  012737  042761  010512          MOV     #42761,SLDSE
                                ;
                                ;SELECT/DESELECT COMMON.
                                ;
010454                          SLDS:
010454  004737  011234          SLDSA:  JSR     PC,GETNAM       ;CHECK NAME. GET MODULE ADDR.
010460  000410                          BR      SLDSC           ;NO NAME. SELECT/DESELECT ALL.
010462  000405                          BR      SLDSL           ;INVALID OR NEX NAME.
010464  013701  001040                  MOV     MODPTR,R1       ;MODULE ADDR TO R1.
010470  042761  040000  000020  SLDSB:  BIC     #BIT14,STAT(R1) ;SELECT/DESELECT MODULE.
010476  000137  007364          SLDSL:  JMP     COMCON          ;DONE.
010502  012702  016236          SLDSC:  MOV     #MODQ,R2        ;MODULE TABLE ADDR TO R2.
010506  012201                  SLDSD:  MOV     (2)+,R1         ;GET MODULE ADDR.
010510  001772                          BEQ     SLDSL           ;BR IF 0.END OF TABLE, DONE.
010512  042761  040000  000020  SLDSE:  BIC     #BIT14,STAT(R1) ;SELECT/DESELECT MODULE.
010520  000772                          BR      SLDSD           ;DO IT AGAIN.
                                ;LOCATIONS SLDSB AND SLDSE ARE PURE WHILE IN RUN MODE.
```

```
 1673  010522                          PON:
 1674  010522                          POFF:
 1675  010522                          KTON:
 1676  010522  000137  007374          KTOFF:  JMP     COMCO3          ;IGNORE THE COMMAND
 1677
 1678                                  ;THIS ROUTINE ENABLES WRITE BUFFER ROTATION
 1679
 1680  010526  105037  001262          ROTON:  CLRB    ROTL            ;CLEAR ROTATION LOCK FLAG
 1681  010532  004737  013262                  JSR     PC,ROTCHK       ;GO DO "ENABLED" MESSAGE
 1682  010536  000137  007374                  JMP     COMCO3          ;CONTINUE
 1683
 1684
 1685                                  ;THIS ROUTINE DISABLES WRITE BUFFER ROTATION
 1686
 1687  010542  112737  000377  001262  ROTOFF: MOVB    #377,ROTL       ;SET THE ROTATION LOCK FLAG
 1688  010550  013737  001266  001264          MOV     ROTNUM,ROTCNT   ;LOAD THE ROTATION COUNTER
 1689  010556  104406  014700                  MSG$,ROTDIS             ;"WBUF ROTATION DISABLED"
 1690  010562  000137  007374                  JMP     COMCO3          ;CONTINUE
```



```
 1691                                  ;MODIFY ROUTINE.
 1692                                  ;GETS ADDRESS AND PRINTS ADDR CONTENTS.
 1693                                  ;USER CAN INPUT NEW NUMBER OR CR.
 1694                                  ;CR MEANS HE JUST WANTS TO EXAMINE.
 1695
 1696  010566  112737  177777  001272  FILL:   MOVB    #-1,FILLID      ;SET FILL INDICATOR.
 1697  010574                          MOD:
 1698  010574  004737  011234          5$:     JSR     PC,GETNAM       ;GET NAME AND MODULE ADDR.
 1699  010600  000401                          BR      4$              ;NO NAME.
 1700  010602  000440                          BR      3$              ;INVALID OR NEX NAME.
 1701  010604  004737  011022          4$:     JSR     PC,GETADR       ;GET ADDRESS.
 1702  010610  000430                          BR      6$              ;INVALID ADDR OR NO ADDR.
 1703  010612  013704  001042                  MOV     ADDRLO,R4               ;MOVE ADDR TO R4.
 1704  010616  063704  001040                  ADD     MODPTR,R4       ;ADD MODULE ADDR TO ADDR.
 1705  010622  010446                  1$:     MOV     R4,-(SP)                ;TYPE ADDR.
 1706  010624  004737  011000                  JSR     PC,ITOA         ;
 1707  010630  011446                  7$:     MOV     (R4),-(SP)      ;GET CONTENTS OF ADDR.
 1708  010632  004737  011000                  JSR     PC,ITOA         ;PRINT IT OUT.
 1709  010636  004737  007404                  JSR     PC,SINPUT       ;GET NEW CONTENTS.
 1710  010642  004737  011070                  JSR     PC,GETNUM       ;CONVERT INPUT TO BINARY.
 1711  010646  000416                          BR      3$              ;INVALID DATA (NOT A NUMBER).
 1712  010650  005737  001062                  TST     YES             ;NUMBER?
 1713  010654  001401                          BEQ     2$              ;BR IF NOT.
 1714  010656  010114                          MOV     R1,(R4)         ;STORE NEW VALUE.
 1715  010660  122702  000012          2$:     CMPB    #12,R2          ;LINE FEED?
 1716  010664  001007                          BNE     3$              ;NO, DONE.
 1717  010666  005724                          TST     (R4)+           ;POINT TO NEXT LOCATION.
 1718  010670  000754                          BR      1$              ;REPEAT FOR NEXT LOC.
 1719  010672  012704  001254          6$:     MOV     #FILCNT,R4      ;GET FILCNT ADDR.
 1720  010676  105737  001272                  TSTB    FILLID          ;IN FILL MODE?
 1721  010702  001352                          BNE     7$              ;BR IF YES.
 1722  010704  000137  007364          3$:     JMP     COMCON          ;DONE.
```

```
1723                                    ;BINARY TO ASCII TYPE ROUTINE. TYPES NUMBER(S) ON STACK.
1724
1725  010710  016646  000002          ITOAX:  MOV     2(SP),-(SP)     ;LOAD EA BITS
1726  010714  016666  000002  000004          MOV     2(SP),4(SP)
1727  010722  016666  000006  000002          MOV     6(SP),2(SP)
1728  010730  004537  011524                  JSR     R5,OACNVX
1729  010734  015515                          HOLD6                   ;AND PUT IT HERE
1730  010736  104406  015515                  MSG$,HOLD6              ;TYPE OUT THE VALUE
1731  010742  000425                          BR      ITOUT
1732
1733  010744  016646  000002          ITOA8:  MOV     2(SP),-(SP)     ;LOAD UPPER BITS
1734  010750  016666  000002  000004          MOV     2(SP),4(SP)     ;AND LOWER BITS
1735  010756  016666  000006  000002          MOV     6(SP),2(SP)     ;AND ADJUST THE STACK
```



```
1736  010764  004537  011532                  JSR     R5,OACNV8       ;CONVERT 8 OCTAL NUMBERS TO ASCII
1737  010770  015526                          HOLD8                   ;AND PUT THEM HERE
1738  010772  104406  015526                  MSG$,HOLD8              ;TYPE OUT THE OCTAL VALUE
1739  010776  000407                          BR      ITOUT           ;GO TYPE IT OUT
1740
1741
1742  011000  016646  000002          ITOA:   MOV     2(SP),-(SP)
1743  011004  004537  011512                  JSR     R5,OACNV        ;CONVERT NUMBER TO ASCII.
1744  011010  015515                          HOLD6
1745  011012  104406  015515                  MSG$,HOLD6              ;TYPE OCTAL VALUE.
1746  011016  012616                  ITOUT:  MOV     (SP)+,(SP)
1747  011020  000207                          RTS     PC              ;EXIT.
```

```
                                ;THIS ROUTINE GETS THE ADDRESS ASSOCIATED WITH THE MOD COMMAND

011022  004737  011070          GETADR: JSR     PC,GETNUM       ;GET NUMBER
011026  000414                          BR      1$              ;INVALID DATA RETURN.
011030  005737  001062                  TST     YES             ;NUMBER?
011034  001411                          BEQ     1$              ;BR IF NOT.
011036  006201                          ASR     R1              ;GOOD ADDRESS?
011040  103410                          BCS     ADRERR          ;NOGOOD, IT'S ODD
011042  060101                          ADD     R1,R1           ;RESTORE THE ADR
011044  010137  001042                  MOV     R1,ADDRLO       ;STORE AT ADDRLO
011050  010337  001044                  MOV     R3,ADDRHI       ;AND ANY HIGH ORDER BITS IN ADDRHI
011054  062716  000002                  ADD     #2,(SP)         ;SET UP GOOD EXIT.
011060  000207                  1$:     RTS     PC              ;DONE

011062  104406  014364          ADRERR: MSG$,INVADR             ;TYPE INVALID ADDR OR DATA.
011066  000207                          RTS     PC              ;EXIT.
```



```
                                ;GET NUMBER FROM KEYBOARD BUFFER, AND RETURN THE LOW ORDER BITS
                                ;IN R1 AND THE HIGH ORDER BITS IN R3.
                                ;ROUTINE ADDS 2 TO THE RETURN PC IF SUCCESSFUL.

011070  013700  001036          GETNUM: MOV     KBPTR,R0        ;GET BUFFER POINTER.
011074  005001                  3$:     CLR     R1              ;DATA
011076  005003                          CLR     R3
011100  005037  001062                  CLR     YES             ;FLAG
011104  112002                  1$:     MOVB    (R0)+,R2        ;GET A BYTE
011106  120227  000040                  CMPB    R2,#40          ;SPACE?
011112  001770                          BEQ     3$              ;YES,IGNORE IT
011114  122702  000015                  CMPB    #15,R2          ;CR?
011120  001425                          BEQ     2$              ;YES,RETURN
011122  120227  000012                  CMPB    R2,#12          ;LF?
011126  001422                          BEQ     2$              ;DONE
011130  120227  000060                  CMPB    R2,#'0          ;LOW LIMIT
011134  002752                          BLT     ADRERR          ;TOO LOW, REPORT THE ERROR
011136  120227  000067                  CMPB    R2,#'7          ;HIGH LIMIT
011142  003347                          BGT     ADRERR          ;TOO HIGH, REPORT THE ERROR
011144  006301                          ASL     R1              ;SHIFT OLD STUFF
011146  006103                          ROL     R3
011150  006301                          ASL     R1              ;3 TIMES LEFT
011152  006103                          ROL     R3
011154  006301                          ASL     R1              ;I.E. MULT. BY OCTAL 10
011156  006103                          ROL     R3
011160  060201                          ADD     R2,R1           ;ADD NEW TO OLD
011162  162701  000060                  SUB     #'0,R1          ;BUT GET RID OF ASCII STUFF
011166  005237  001062                  INC     YES             ;SET FLAG
011172  000744                          BR      1$              ;MORE, MORE
011174  010037  001036          2$:     MOV     R0,KBPTR        ;SAVE CURRENT BUF POINTER.
011200  062716  000002                  ADD     #2,(SP)         ;SET UP OK EXIT.
011204  000207                          RTS     PC              ;EXIT.
```

```
                                ;THIS ROUTINE PUTS THE MODULE NAME INTO AN ASCII STRING FOR OUTPUT
                                ;
011206  010346          FILLNM: MOV     R3,-(SP)                ;SAVE R3.
011210  012503                  MOV     (R5)+,R3                ;STORE ADDR TO R3.
011212  012704  000005          MOV     #5,R4           ;WILL DO 5 TIMES.
011216  112123          1$:     MOVB    (R1)+,(R3)+     ;MOVE CHAR.
011220  005304                  DEC     R4              ;DONE?
011222  001375                  BNE     1$              ;BR IF NOT.
011224  162701  000005          SUB     #5,R1           ;RESTORE R1.
011230  012603                  MOV     (SP)+,R3                ;RESTORE R3.
011232  000205                  RTS     R5              ;EXIT.
```



```
                                ;THIS POUTINE GETS THE MODULE NAME ASSOCIATED WITH THE MOD AND SEL/DES COMMANDS.
                                ;
011234  013700  001036  GETNAM: MOV     KBPTR,R0        ;GET BUFFER POINTER.
011240  005037  001062          CLR     YES             ;CLEAR SUCCESS INDICATOR.
011244  005037  001040          CLR     MODPTR          ;IF 0, ABS ADDR.
011250  012702  000005          MOV     #5,R2           ;WILL DO 5 TIMES.
011254  012701  015415          MOV     #AMODNM+1,R1    ;WILL STORE KYBD NAME IN TMPNAM.
011260  112721  000040  1$:     MOVB    #40,(R1)+       ;BUT CLEAR IT TO SPACES FIRST.
011264  005302                  DEC     R2              ;DONE?
011266  001374                  BNE     1$              ;BR IF NOT.
011270  012701  015415          MOV     #AMODNM+1,R1    ;RELOAD TEMP ADDR.
011274  111002          2$:     MOVB    (R0),R2         ;GET CHAR FROM KBUF AREA.
011276  122702  000015          CMPB    #15,R2          ;CR?
011302  001442                  BEQ     6$              ;BR IF YES.
011304  122702  000012          CMPB    #12,R2          ;LF? TREAT AS CR.
011310  001437                  BEQ     6$              ;BR IF YES.
011312  122702  000040          CMPB    #40,R2          ;SPACE? DISREGARD LEADING ONES.
011316  001430                  BEQ     5$              ;BR IF YES.
011320  120227  000101          CMPB    R2,#'A          ;A OR HIGHER?
011324  002403                  BLT     3$              ;BR IF LESS. (COULD BE A NUMBER).
011326  120227  000132          CMPB    R2,#'Z          ;Z OR LESS?
011332  003411                  BLE     4$              ;BR IF Z OR LESS.
011334  120227  000060  3$:     CMPB    R2,#'0          ;0 OR HIGHER?
011340  002427                  BLT     7$              ;BR IF LESS. YOU LOOSE!
011342  120227  000071          CMPB    R2,#'9          ;9 OR LESS?
011346  003024                  BGT     7$              ;BR IF GREATER. YOU LOOSE!.
011350  005737  001062          TST     YES             ;ANY PREVIOUS CHARS?
011354  001426                  BEQ     8$              ;BR IF NOT. NOT LEADING NUMERICS ALLOWED!.
011356  023727  001062  000005  4$:     CMP     YES,#5          ;6TH CHARACTER?
011364  002015                  BGE     7$              ;IF YES, BRANCH
011366  110221                  MOVB    R2,(R1)+                ;STORE CHAR.
011370  005237  001062          INC     YES
011374  005200          41$:    INC     R0              ;POINT TO NEXT CHAR.
011376  000736                  BR      2$              ;DO IT AGAIN!
011400  005737  001062  5$:     TST     YES             ;ANY CHARS?


011404  001773                  BEQ     41$             ;BR IF NOT.
011406  000416                  BR      9$

011410  005737  001062  6$:     TST     YES             ;ANY CHARS?
011414  001013                  BNE     9$              ;BR IF YES.
011416  000405                  BR      8$              ;NO NAME.
011420  104406  014403  7$:     MSG$,INVNAM             ;TYPE INVALID OR NEX (NON-EXISTENT) MESSAGE.
011424  062716  000002          ADD     #2,(SP)         ;SET UP INVALID NAME EXIT.
011430  000207                  RTS     PC              ;EXIT.


011432  005037  001062  8$:     CLR     YES             ;
011436  010037  001036          MOV     R0,KBPTR        ;SAVE CURRENT BUF POINTER.
011442  000207                  RTS     PC              ;EXIT.
011444  012702  016236  9$:     MOV     #MODQ,R2        ;MODULE TABLE ADDR TO R2.
011450  012203  10$:    MOV     (R2)+,R3                ;GET MODULE ADDR.
011452  001762                  BEQ     7$              ;INVALID NAME, NO MODULE!!
011454  012701  015415          MOV     #AMODNM+1,R1
```

```
 1863  011460  012704  000005                MOV     #5,R4           ;WILL MATCH 5 CHARS.
 1864  011464  122123                  11$:  CMPB    (R1)+,(3)+      ;CHAR MATCH?
 1865  011466  001370                        BNE     10$             ;BR IF NOT.
 1866  011470  005304                        DEC     R4              ;DONE?
 1867  011472  001374                        BNE     11$             ;BR IF NOT.
 1868  011474  014237  001040                MOV     -(R2),MODPTR    ;MODULE ADDR TO MODPTR.
 1869  011500  010037  001036                MOV     R0,KBPTR        ;SAVE CURRENT BUF POINTER.
 1870  011504  062716  000004                ADD     #4,(SP)         ;SET UP SUCCESS EXIT.
 1871  011510  000207                        RTS     PC              ;EXIT.
```



```
 1872                                  ;OCTAL TO ASCII CONVERT ROUTINE.
 1873                                  ;CONVERTS ARGUMENT ON STACK TO ASCII, STORES STRING STARTING AT ADDRESS
 1874                                  ;GIVEN FOLLOWING THE CALL
 1875                                  ;EXTENDED ENTRY REQUIRES ADDITIONAL  ARGUMENT ON STACK= AFTER
 1876                                  ;PUSHING LOWER 16 BITS ONTO STACK, PUSH EA BITS (SHIFTED TO POS 4,5)
 1877
 1878  011512  005037  001112          OACNV:  CLR     OASCEA          ;CLEAR OUT UPPER BITS -- NOT USED
 1879  011516  105037  001310                  CLRB    ATE             ;CLEAR 8 CHARACTER FLAG
 1880  011522  000412                        BR      CONVRT          ;CONTINUE
 1881
 1882  011524  105037  001310          OACNVX: CLRB    ATE             ;CLEAR 8 CHARACTER FLAG
 1883  011530  000403                        BR      HOP             ;CONTINUE
 1884
 1885  011532  112737  000200  001310  OACNVR: MOVB    #BIT7,ATE       ;SET 8 CHARACTER FLAG
 1886
 1887  011540  016637  000002  001112  HOP:    MOV     2(SP),OASCEA
 1888  011546  012616                        MOV     (SP)+,(SP)
 1889  011550  004437  012244          CONVRT: JSR     R4,SAV04        ;SAVE REGS 0-4.
 1890  011554  016600  000014                  MOV     12.(SP),R0      ;GET OCTAL VALUE.
 1891  011560  012501                        MOV     (R5)+,R1                ;GET DEST ADDR.
 1892  011562  012702  000006                MOV     #6,R2           ;SET CONVERT COUNT TO 6.
 1893  011566  060201                        ADD     R2,R1           ;DEVELOP ADDR TO STORE 1ST CHAR.
 1894  011570  105737  001310                TSTB    ATE             ;SET UP TO DO 8 CHARACTERS ?
 1895  011574  100002                        BPL     3$              ;NO, CONTINUE
 1896  011576  062701  000002                ADD     #2,R1           ;YES, ADD 2 TO THE ADDRESS
 1897  011602  010003                  3$:   MOV     R0,R3           ;GET VALUE TO R3.
 1898  011604  042703  177770                BIC     #177770,R3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
 1899  011610  062703  000060                ADD     #60,R3          ;CONVERT TO ASCII.
 1900  011614  110341                        MOVB    R3,-(R1)                ;STORE IT.
 1901  011616  042700  000007                BIC     #7,R0           ;CLEAR DIGIT JUST CONVERTED.
 1902  011622  006000                        ROR     R0              ;SHIFT IN NEXT DIGIT.
 1903  011624  006000                        ROR     R0
 1904  011626  006000                        ROR     R0
 1905  011630  005302                        DEC     R2              ;DONE 6 DIGITS?
 1906  011632  001363                        BNE     3$              ;BR IF NOT.
 1907  011634  105737  001310                TSTB    ATE             ;SET UP TO DO 8 CHARACTERS ?
 1908  011640  100410                        BMI     4$              ;YES, DO IT
 1909  011642  013702  001112                MOV     OASCEA,R2
 1910  011646  006202                        ASR     R2
 1911  011650  006202                        ASR     R2
 1912  011652  006202                        ASR     R2
 1913  011654  060203                        ADD     R2,R3
 1914  011656  110311                        MOVB    R3,(R1)
 1915  011660  000425                        BR      5$              ;GO FINISH UP
 1916
 1917  011662  132737  000001  001310  4$:   BITB    #BIT0,ATE       ;ALL DONE ?
 1918  011670  001021                        BNE     5$              ;YES, GET OUT
 1919  011672  013700  001112                MOV     OASCEA,R0       ;NO, LOAD UPPER NUMBERS TO BE CONVERTED
 1920  011676  006300                        ASL     R0              ;LINE UP BITS BELONGING TO LOWER NUMBER
 1921  011700  042700  177770                BIC     #177770,R0      ;GIT RID OF ALL OTHERS
 1922  011704  060003                        ADD     R0,R3           ;ADD IT IN TO COMPLETE THE NUMBER
 1923  011706  110311                        MOVB    R3,(R1)         ;STORE IT
 1924  011710  013700  001112                MOV     OASCEA,R0       ;LOAD UPPER BITS AGAIN
 1925  011714  006200                        ASR     R0              ;GIT RID OF THE 2 BITS
 1926  011716  006200                        ASR     R0              ;JUST TAKEN CARE OF
 1927  011720  012702  000002                MOV     #2,R2           ;SET UP TO DO THE LAST 2 CHARACTERS
```

```
 1928  011724  152737  000001  001310          BISB    #BIT0,ATE       ;SET 'ALL DONE' FLAG
 1929  011732  000723                          BR      3$              ;GO DO THE LAST 2 CHARACTERS
 1930
 1931  011734  004737  012256          5$:     JSR     PC,RST04        ;RESTORE REGS. 0 THRU 4
 1932  011740  012616                          MOV     (SP)+,(SP)      ;
 1933  011742  000205                          RTS     R5              ;RETURN
```



```
 1934                                  ;BINARY TO DECIMAL ASCII CONVERT ROUTINE.
 1935                                  ;CONVERTS ARGUMENT ON STACK TO DECIMAL ASCII, STORES IT
 1936                                  ;STARTING AT ADDRESS GIVEN AFTER THE CALL.
 1937                                  ;REMOVES ARGUMENT FROM STACK BEFORE RETURNING
 1938
 1939  011744  004437  012244          BDCNV:  JSR     R4,SAV04        ;SAVE REGS 0-4.
 1940  011750  016601  000014                  MOV     12.(SP),R1      ;GET BIN VALUE.
 1941  011754  012700  016230                  MOV     #DECVAL,R0      ;GET ADDR OF DECVAL STRING.
 1942  011760  012702  012044                  MOV     #TENPWR,R2      ;ADDR OF TENPWR TO R2.
 1943  011764  012703  000005                  MOV     #5,R3           ;SET UP TO DO 5 CONVERSIONS.
 1944  011770  005004                  1$:     CLR     R4              ;CLEAR RESULT.
 1945  011772  161201                  2$:     SUB     (R2),R1         ;SUBTRACT TEN POWER.
 1946  011774  103402                          BCS     3$              ;BR IF UNSUCCESSFUL.
 1947  011776  005204                          INC     R4              ;ADD 1 TO RESULT.
 1948  012000  000774                          BR      2$              ;DO IT AGAIN.
 1949  012002  062201                  3$:     ADD     (R2)+,R1                ;RESTORE SUBTRACTED VALUE.
 1950  012004  062704  000060                  ADD     #60,R4          ;MAKE IT ASCII.
 1951  012010  110420                          MOVB    R4,(R0)+                ;SAVE IT.
 1952  012012  005303                          DEC     R3              ;DONE 5?
 1953  012014  001365                          BNE     1$              ;BR IF NOT.
 1954  012016  012501                          MOV     (R5)+,R1                ;GET FINAL STORE ADDR.
 1955  012020  012702  000005                  MOV     #5,R2           ;GET DIGIT COUNT DESIRED.
 1956  012024  060201                          ADD     R2,R1           ;COMPUTE ADDR OF 1ST DIGIT.
 1957  012026  114041                  4$:     MOVB    -(R0),-(R1)     ;TRANSFER CHAR.
 1958  012030  005302                          DEC     R2              ;DONE?
 1959  012032  001375                          BNE     4$              ;BR IF NOT.
 1960  012034  004737  012256                  JSR     PC,RST04        ;RESTORE REGS 0-4.
 1961  012040  012616                          MOV     (SP)+,(SP)
 1962  012042  000205                          RTS     R5              ;EXIT.
 1963  012044  023420  001750  000144  TENPWR: 10000.,1000.,100.,10.,1
 1964  012052  000012  000001
```

```
                                        ;RETURN TO KERNEL MODE
                                        ;
012056  042766  170000  000002  RTTK,:  BIC     #170000,2(SP)
012064  000002                          RTI
                                        ;
                                        ;
                                        ; ROUTINE TO CALC NEXT RANDOM NUMBER, USES A 29 BIT SHIFT REGISTER
                                        ; ALGORITHM, EXCLUSIVELY OR-ING THE 28TH AND 1ST(LOW ORDER) BITS
                                        ; INTO THE 29TH BIT AS IT SHIFTS RIGHT, SHIFTS 16 TIMES EACH TIME
                                        ; ITS CALLED.  MODULES DO A RAND CALL THEN JUST REFERENCE RANNUM
                                        ; WHICH IS DEFINED AS GLOBAL IN DDXCOM
                                        ;
012066  004437  012244          RAND,:  JSP     R4,SAV04        ;SAVE THE REGISTERS
012072  013700  001162                  MOV     RANWRK,R0       ;GET UPPER BITS
012076  013701  001160                  MOV     RANNUM,R1       ;GET LOWER ORDER 16 BITS
012102  012702  000020                  MOV     #16.,R2         ;SHIFT 16 TIMES
012106  006000          RLOOP:  ROR     R0              ;SHIFT R0 INTO CARRY
012110  006001                  ROR     R1              ;SHIFT CARRY INTO R1, R1 INTO CARRY
012112  103410                  BCS     1$              ;FIRST HALF OF EX-OR PROCESS

012114  042700  010000          BIC     #10000,R0       ;ASSUME OLD BIT 28 WAS ALSO CLEAR
012120  032700  002000          BIT     #2000,R0        ;CHECK THAT ASSUMPTION
012124  001412                  BEQ     2$              ;BR IF RIGHT, ELSE
012126  052700  010000          BIS     #10000,R0       ;PUT BIT IN
012132  000407                  BR      2$              ;GO SHIFT AGAIN

012134  052700  010000  1$:     BIS     #10000,R0       ;ASSUME OLD BIT 28 WASN'T ALSO SET
012140  032700  002000          BIT     #2000,R0        ;CHECK THAT ASSUMPTION
012144  001402                  BEQ     2$              ;BR IF RIGHT
012146  042700  010000          BIC     #10000,R0       ;ELSE 2 1'S = A ZERO BIT

012152  005302          2$:     DEC     R2              ;DONE 16 TIMES?
012154  001354                  BNE     RLOOP           ;BR IF NO
012156  010037  001162          MOV     R0,RANWRK       ;SAVE CURRENT VALUES
012162  010137  001160          MOV     R1,RANNUM
012166  004737  012256          JSR     PC,RST04        ;RESTORE REG'S
012172  000002                  RTI                     ;RETURN FROM TRAP CALL
```



```
                                        ;SUBROUTINE TO GET PAGE ADDRESS FIELD VALUE (TOP 12 BITS OF ADDRESS)
                                        ;CALL VIA:
                                        ;       JSR R5,TOP12S
                                        ;       ADDRESS OF TABLE CONTAINING LOWER 16 BITS, TOP 2 BITS (SHIFTED),
                                        ;       AND SLOT FOR RETURNED VALUE OF ADDRESS

012174  010046          TOP12S: MOV     R0,-(SP)        ;SAVE THE OLD VALUE OF R0
012176  012500                  MOV     (R5)+,R0        ;GET THE ADDRESS OF THE TABLE-PUT IT IN R0
012200  012046                  MOV     (R0)+,-(SP)     ;FETCH THE ADDRESS TO BE ACTED ON TO THE STACK
012202  006216                  ASR     (SP)            ;ELIMINATE THE SIX LOW ORDER BITS, THIS ONE DROPS BIT0
012204  006216                  ASR     (SP)            ;THIS ROTATION DROPS BIT1
012206  006216                  ASR     (SP)            ;THIS ROTATION DROPS BIT2
012210  006216                  ASR     (SP)            ;THIS ROTATION DROPS BIT3
012212  006216                  ASR     (SP)            ;THIS ROTATION DROPS BIT 4
012214  006216                  ASR     (SP)            ;AND THIS ROTATION DROPS BIT5
012216  042716  176000          BIC     #176000,(SP)    ;ISOLATE THE 12 HIGH ORDER BITS IN THE LOWER
012222  011660  000002          MOV     (SP),2(R0)      ;SAVE THE 12 HIGH ORDER BITS IN THE TABLE'S 3RD ENTRY
012226  012016                  MOV     (R0)+,(SP)      ;FETCH THE EXTENDED ADDRESS BITS FROM THE 2ND ENTRY
012230  000316                  SWAB    (SP)            ;MOVE THE BITS TO BIT POSITIONS 6+7,FIRST, TO 8+9,
012232  006216                  ASR     (SP)            ;THEN MOVE THEM TO 7 + 8
012234  006216                  ASR     (SP)            ;FINALLY ROTATE THEM TO 6 + 7
012236  052610                  BIS     (SP)+,(R0)      ;ADD THE EXTENDED ADDRESS BITS TO THE 3RD ENTRY
012240  012600                  MOV     (SP)+,R0        ;RESTORE THE OLD VALUE OF R0
012242  000205                  RTS     R5              ;RETURN TO THE CALLING PROCEDURE
```

```
                                ;SAVE REGS 0-4 SUBROUTINE, (CALL VIA JSR R4,SAVQ4)
                                
012244  010346                  SAV04:  MOV     R3,-(SP)        ;SAVE R3
012246  010246                          MOV     R2,-(SP)        ;SAVE R2
012250  010146                          MOV     R1,-(SP)        ;SAVE R1
012252  010046                          MOV     R0,-(SP)        ;SAVE R0
012254  010407                          MOV     R4,PC           ;R4 IS ALREADY SAVED
                                
                                ;RESTORE REGS 0-4 SUBROUTINE, (CALL VIA JSR PC,RST04)
                                
012256  012604                  RST04:  MOV     (SP)+,R4        ;RETURN ADDRESS
012260  012600                          MOV     (SP)+,R0        ;RESTORE R0
012262  012601                          MOV     (SP)+,R1        ;R1
012264  012602                          MOV     (SP)+,R2        ;R2
012266  012603                          MOV     (SP)+,R3
012270  000204                          RTS     R4              ;RESTORE R4 AND RETURN
```



```
                                ;CLEAR VARIABLES AND QUEUES, AND LOAD TRAP CATCHER STARTING AT 70
                                
012272  012700  001006          CLRQUS: MOV     #IOQUE,R0       ;CLEAR VARIABLES.
012276  012701  000140                  MOV     #TKS-IOQUE,R1
012302  105020                  1$:     CLRB    (R0)+
012304  005301                          DEC     R1
012306  001375                          BNE     1$
012310  012700  016606                  MOV     #IOQ,R0         ;INITIALIZE QUEUE POINTERS
012314  010037  001074                  MOV     R0,IOQ1
012320  010037  001076                  MOV     R0,IOQ2
012324  012737  017116  001100          MOV     #TYPEQ,TYPQ1
012332  012737  017116  001102          MOV     #TYPEQ,TYPQ2
012340  012701  000764                  MOV     #IOQL+TYPQL,R1  ;SET UP TO CLEAR QUEUES.
012344  105020                  2$:     CLRB    (R0)+
012346  005301                          DEC     R1              ;DONE?
012350  001375                          BNE     2$              ;BR IF NOT.
012352  000207                          RTS     PC
```

```
                                ;LOAD TRAPCATCHER IN VECTOR AREA

012354  012700  000072  CLRVEC: MOV     #72,R0          ;FILL VECTOR AREA WITH .+2
012360  012701  000070          MOV     #70,R1          ;AND HALT.
012364  010021          1$:     MOV     R0,(R1)+
012366  005021                  CLR     (R1)+
012370  010100                  MOV     R1,R0
012372  005720                  TST     (R0)+
012374  020027  001002          CMP     R0,#1002        ;FILLED UP?
012400  001371                  BNE     1$              ;BR IF NOT.
012402  000207                  RTS     PC              ;YES. EXIT.

                                ;TRAP INTERPRETER ROUTINE.

012404  010046  TRPINT: MOV     R0,-(SP)                ;PUSH R0.
012406  016600  000002          MOV     2(SP),R0        ;GET TRAP PC.
012412  014000                  MOV     -(R0),R0        ;GET TRAP CALL.
012414  006300                  ASL     R0              ;MULTIPLY BY 2.
012416  062700  001446          ADD     #TRPTAB-TRP2,R0 ;FORM TABLE ADDRESS
012422  020027  012516          CMP     R0,#TRPLIM      ;WITHIN LIMITS?
012426  103002                  BHIS    1$              ;BR IF NOT.
012430  011000                  MOV     @R0,R0          ;GET ROUTINE ADDRESS
012432  000200                  RTS     R0              ;GO TO ROUTINE. RESTORE R0.
012434  104404  014223  1$:     MSG$;INVTRP     ;ERROR INVALID TRAP CALL. CRASH SYSTEM.
012440  104406  014620          MSG$,HLT        ;'HALT'
012444  000000                  HALT
        011000                  TRP2=11000
        000000                  TRAPX=0
012446                  TRPTAB:
012446                          TRPDEF  EXIT$,EXIT.
012446  005030                  EXIT.                   ;POINTER FOR TRAP CALL EXIT$
012450                          TRPDEF  QUE$,QUE.
012450  005470                  QUE.                    ;POINTER FOR TRAP CALL QUE$
012452                          TRPDEF  ENDPS$,TYPQ.
012452  005324                  TYPQ.                   ;POINTER FOR TRAP CALL ENDPS$
012454                          TRPDEF  END$,TYPQ.
012454  005324                  TYPQ.                   ;POINTER FOR TRAP CALL END$
012456                          TRPDEF  ERROR$,TYPQ2.
012456  005250                  TYPQ2.                  ;POINTER FOR TRAP CALL ERROR$
012460                          TRPDEF  DATER$,TYPQ2.
012460  005250                  TYPQ2.                  ;POINTER FOR TRAP CALL DATER$
012462                          TRPDEF  MSG$,TYPQ1.
012462  005336                  TYPQ1.                  ;POINTER FOR TRAP CALL MSG$
012464                          TRPDEF  BREAK$,LDBRK.
012464  005460                  LDBRK.                  ;POINTER FOR TRAP CALL BREAK$
012466                          TRPDEF  ERRN$,TYPQ2.
012466  005250                  TYPQ2.                  ;POINTER FOR TRAP CALL ERRN$
012470                          TRPDEF  MSGN$,TYPQ.
012470  005324                  TYPQ.                   ;POINTER FOR TRAP CALL MSGN$
012472                          TRPDEF  GWBUF$,GWBUF.
012472  013646                  GWBUF.                  ;POINTER FOR TRAP CALL GWBUF$
012474                          TRPDEF  GETPA$,GETPA.
012474  013770                  GETPA.                  ;POINTER FOR TRAP CALL GETPA$
012476                          TRPDEF  CDATA$,CDATA.
012476  006626                  CDATA.                  ;POINTER FOR TRAP CALL CDATA$
012500                          TRPDEF  MAP22$,MAP22.
```



```
012500  014012                  MAP22.                  ;POINTER FOR TRAP CALL MAP22$
012502                          TRPDEF  MSGS$,TYPQ.
012502  005324                  TYPQ.                   ;POINTER FOR TRAP CALL MSGS$
012504                          TRPDEF  DATCK$,DATCK.
012504  006616                  DATCK.                  ;POINTER FOR TRAP CALL DATCK$
012506                          TRPDEF  RAND$,RAND.
012506  012066                  RAND.                   ;POINTER FOR TRAP CALL RAND$
012510                          TRPDEF  DERRX$,TYPQ3.
012510  005150                  TYPQ3.                  ;POINTER FOR TRAP CALL DERRX$
012512                          TRPDEF  MSGD$,TYPQ4.
012512  005102                  TYPQ4.                  ;POINTER FOR TRAP CALL MSGD$
012514                          TRPDEF  RTTK$,RTTK.
012514  012056                  RTTK.                   ;POINTER FOR TRAP CALL RTTK$
012516                  TRPLIM:
```

```
2130                                    ;BUS ERROR AND RESERVED INSTRUCTION TRAP ROUTINES
2131                                    ;CHECK MODE BITS OF PSW ON STACK (MAYBE USER IF KT11 IN USE)
2132                                    ;ORDER OF ARGUMENTS: VIRTUAL PC ON STACK, PSW ON STACK, STACK POINTER
2133                                    ;AFTER TRAP, VIRTUAL ADDRESS TRAPPED TO
2134
2135  012516  012746  000004          BUSERR: MOV     #4,-(SP)                ;INDICATE BUS ERROR TRAP.
2136  012522  000402                          BR      RESIA
2137  012524  012746  000010          RESINT: MOV     #10,-(SP)       ;INDICATE RESERVED INSTRUCTION TRAP.
2138  012530  000240                  RESIA:  NOP
2139  012532  010605                          MOV     SP,R5           ;SAVE THE STACK POINTER.
2140  012534  012706  016606                  MOV     #SPSPEC,SP      ;PUT INFORMATION INTO SPECIAL MONITOR STACK
2141  012540  012546                          MOV     (R5)+,-(SP)     ;PUSH TRAP VECTOR ON STACK
2142  012542  010546                          MOV     R5,-(SP)        ;PUSH STACK POINTER ON STACK
2143  012544  016546  000002                  MOV     2(R5),-(SP)     ;PUSH THE PSW ON STACK
2144  012550  011546                          MOV     (R5),-(SP)      ;PUSH THE PC ON STACK
2145  012552  012706  016546                  MOV     #SPBOT,SP       ;RESTORE MONITOR STACK POINTER
2146  012556  004737  006230                  JSR     PC,CTRLX        ;CLEAR QUEUES, CLEAN UP, MAP TO BANKS 0+1
2147  012562  105237  001244                  INCB    SYSCNT          ;COUNT A SYSTEM ERROR
2148  012566  012706  016576                  MOV     #SPSPEC-10,SP   ;SETUP SPECIAL STACK POINTER
2149  012572  005037  001060                  CLR     IOBKID          ;ALLOW TYPING
2150  012576  112737  000001  001013          MOVB    #1,BRAKE        ;PREVENT MODULE STARTUP DURING MESSAGE
2151  012604  104406  014512                  MSG$,SYSERR             ;TYPE SYS ERROR FAILURE.
2152  012610  012705  011000                  MOV     #ITOA,R5
2153  012614  004715                          JSR     PC,(R5)         ;TYPE ERR PC.
2154  012616  004715                          JSR     PC,(R5)         ;TYPE ERR PSW.
2155  012620  004715                          JSR     PC,(R5)         ;TYPE SP AFTER ERROR
2156  012622  004715                          JSR     PC,(R5)         ;TYPE TRAP VECTOR ADDRESS
2157
2158  012624  012706  016546                  MOV     #SPBOT,SP       ;RESTORE MONITOR STACK POINTER
2159  012630  104406  014122          2$:     MSG$,DOT                ;TYPE CRLF AND DOT
2160  012634  000426                          BR      PWRUA           ;ATTEMPT TO CONTINUE
```



```
2161                                    ;POWER DOWN AND POWER UP ROUTINES.
2162
2163  012636                    PWRDN:
2164  012636  012737  012646  000024  1$:     MOV     #PWRUP,PWRFV    ;SET UP POWER UP VECTOR.
2165  012644  000000                          HALT                    ;POWER DOWN HALT
2166  012646  012737  012636  000024  PWRUP:  MOV     #PWRDN,PWRFV    ;SET UP POWER DOWN VECTOR.
2167  012654  012706  016546                  MOV     #SPBOT,R6       ;RESET STACK.
2168  012660  005237  001246                  INC     PWRCNT          ;COUNT A POWER FAIL
2169  012664  105037  001274                  CLRB    RUNRFL          ;CLEARRESUME RUN FLAG
2170  012670  004737  006230                  JSR     PC,CTRLX        ;CLEAR QUEUES AND INITIALIZE KT11 IF IN USE.
2171  012674  005037  001060                  CLR     IOBKID          ;ALLOW TYPING
2172  012700  112737  000001  001013          MOVB    #1,BRAKE        ;PREVENT MODULE STARTUP
2173  012706  104406  014554                  MSG$,PWRFAI             ;TYPE POWER FAILURE MESSAGE.
2174  012712                    PWRUA:
2175  012712  105737  001257          3$:     TSTB    RMODE           ;WERE WE IN RUN MODE?
2176  012716  001021                          BNE     LOGICA          ;BR IF YES TO RESTART.
2177  012720  000400                          BR      CHNOUT          ;NO, CONTINUE
```

```
                                        ;ROUTINE TO EXIT TO CHAIN MONITOR, OR RETURN TO KYBD RTN,
                                        ;PROGRAM SHOULD ALWAYS BE IN BANK 0 BY HERE

012722  105737  001260        CHNOUT: TSTB    CHN             ;IN CHAIN MODE?
012726  100402                        BMI     1$              ;BR IF YES,
012730  000137  007364                JMP     COMCON          ;BACK TO KYBD SERVICE,
012734  105737  001252        1$:     TSTB    KTPRES
012740  001402                        BEQ     2$              ;BRANCH IF NO KT11
012742  005037  177572                CLR     @#SSR0          ;TURN OFF KT11
012746  013700  000042        2$:     MOV     42,R0
012752  004710                LOGIC:  JSR     PC,(R0)         ;RETURN TO MONITOR,
012754  000240  000240  000240        .WORD   NOP,NOP,NOP
012762                        LOGICA:
012762  105737  001260        5$:     TSTB    CHN
012766  001402                        BEQ     6$
012770  000137  001414                JMP     START
012774  000137  007476        6$:     JMP     RUNRES
```



```
                                        ;SUBROUTINE TO ROTATE WRITE BUFFERS IF ROTI INDICATOR =1,

013000  105737  001261        ROTBUF: TSTB    ROTI            ;ROTATION ALLOWED?
013004  001525                        BEQ     ROTXT           ;BR IF NOT,
013006  105737  001262                TSTB    ROTL            ;IS ROTATION LOCKED ?
013012  100016                        BPL     1$              ;NO, CONTINUE
013014  105737  001275                TSTB    LOCK            ;YES, IS THE PROGRAM LOCKED ?
013020  001013                        BNE     1$              ;YES, DON'T SET STOP
013022  005337  001264                DEC     ROTCNT          ;NO, TIME TO RELOCATE ?
013026  003114                        BGT     ROTXT           ;NO, CONTINUE
013030  105737  001320                TSTB    NSTOP           ;ALLOW STOP TO SET ?
013034  001002                        BNE     10$             ;NO, CONTINUE
013036  105237  001024                INCB    STOP            ;YES, SET THE STOP FLAG
013042  013737  001266  001264  10$:   MOV     ROTNUM,ROTCNT   ;RESTORE THE COUNTER
013050  105737  001024        1$:     TSTB    STOP            ;IF STOP IS SET, DON'T ROTATE
013054  001101                        BNE     ROTXT
013056  105737  001262                TSTB    ROTL            ;ROTATE LOCKED?
013062  100476                        BMI     ROTXT           ;BR IF YES
013064  105737  001021                TSTB    MDXCTR          ;IF COUNTER NOT INITIALIZED, DON'T ROTATE
013070  100473                        BMI     ROTXT
013072  013746  001212                MOV     WBUF,-(SP)      ;CHECK CURRENT VALUE
013076  042716  160000                BIC     #160000,(SP)
013102  022726  016000                CMP     #16000,(SP)+
013106  003016                        BGT     3$              ;IF NOT LAST IN BANK, BRANCH
013110  105737  001021                TSTB    MDXCTR          ;HAVE ALL MODULES USED A VALUE IN THIS BANK?
013114  001403                        BEQ     2$              ;YES- TIME TO CHANGE BANKS
013116  105337  001302                DECB    $GWBFC          ;NO, DEC TIMEOUT COUNTER
013122  001056                        BNE     ROTXT           ;EXIT UNLESS TIMEOUT HAS OCCURRED
013124  113737  001022  001021  2$:   MOVB    MDXCNT,MDXCTR   ;RESET COUNTERS
013132  113737  001031  001302        MOVB    ROTCT,$GWBFC
013140  105137  001023                COMB    WBFLG           ;TOGGLE INDICATOR
013144  023737  001216  001222  3$:   CMP     WBUF12,OFFSET   ;LOCORE ROTATION?
013152  062737  000020  001216        ADD     #20,WBUF12
013160  023737  001216  001206        CMP     WBUF12,HCOR12   ;CHECK FOR TOP OF WRITE BUFFER AREA
013166  103417                        BLO     ROTB2           ;IF NOT, BRANCH
013170  105737  001263        ROTB1:  TSTB    RELOC           ;RELOCATION ALLOWED?
013174  001426                        BEQ     ROTB3           ;NO- BRANCH
013176  105737  001275                TSTB    LOCK            ;LOCKED?
013202  001023                        BNE     ROTB3           ;YES- BRANCH
013204  105737  001320                TSTB    NSTOP           ;ALLOW STOP TO SET ?
013210  001002                        BNE     1$              ;NO, CONTINUE
013212  105237  001024                INCB    STOP            ;YES, SET STOP
013216  162737  000020  001216  1$:   SUB     #20,WBUF12      ;RESTORE WBUF
013224  000415                        BR      ROTXT           ;EXIT
013226  062737  002000  001212  ROTB2: ADD     #2000,WBUF      ;BUMP ADDR- POINT 512 WORDS HIGHER
013234  103003                        BCC     1$
013236  062737  000020  001214        ADD     #20,WBFEA
013244  004737  010166        1$:     JSR     PC,SETWB1       ;CALCULATE MXWBF
013250  000403                        BR      ROTXT
013252  004737  010126        ROTB3:  JSR     PC,SETWBF
013256  000400                        BR      ROTXT           ;RETURN

013260  000207                ROTXT:  RTS     PC              ;RETURN
```

```
                                ;SUBROUTINE TO CHECK IF BUFFER ROTATION ENABLED, AND TO TYPE
                                ;THE ROTATION RANGE.
                                
013262  105737  001303  ROTCHK: TSTB    MEMERF          ;ANY MEMORY ERRORS?
013266  001406                  BEQ     1$              ;NO- BRANCH
013270  104406  016061          MSG$,MERR1
013274  104406  014576          MSG$,MEMCHG
013300  105037  001303          CLRB    MEMERF
013304  105737  001261  1$:     TSTB    ROTI            ;ROTATE ENABLED?
013310  001444                  BEQ     2$              ;BR IF NOT.
013312  104406  014630          MSG$,ROTENB
013316  013746  001200          MOV     LOCORE,-(SP)    ;TYPE LOWER LIMIT.
013322  004737  011000          JSR     PC,ITOA
013326  005037  001120          CLR     SAVHI           ;ZERO ANY HIGH ORDER BITS
013332  013737  001206  001116  MOV     HCOR12,SAVLO    ;COPY INTO LOW ORDER BITS
013340  006337  001116          ASL     SAVLO           ;MAKE IT A 16-BIT NUMBER
013344  006337  001116          ASL     SAVLO           ;
013350  006337  001116          ASL     SAVLO           ;
013354  006337  001116          ASL     SAVLO           ;
013360  006337  001116          ASL     SAVLO           ;
013364  006337  001116          ASL     SAVLO           ;
013370  162737  000002  001116  SUB     #2,SAVLO
013376  103003                  BCC     3$
013400  162737  000020  001120  SUB     #20,SAVHI
013406  013746  001116  3$:     MOV     SAVLO,-(SP)     ;TYPE UPPER LIMIT
013412  013746  001120          MOV     SAVHI,-(SP)
013416  004737  010710          JSR     PC,ITOAX
                                
013422  000207          2$:     RTS     PC              ;EXIT.
```



```
                                ;       THIS ROUTINE WILL CONVERT THE GIVEN BINARY
                                ;NUMBER INTO HOURS, MINUTES, AND SECONDS.  LOCATIONS
                                ;HOLDING THE FINAL NUMBERS WILL BE TAGGED AS:  HOURS,
                                ;MINITS, AND SEKNDS. HOURS MUST BE ON AN ODD BOUNDARY.
                                ;MINITS AND SEKNDS MUST BE ON EVEN BOUNDARIES.
                                ;TEMPORARY LOCATION HOLD5 MUST BE ON AN ODD BOUNDARY.
                                ;               CALL VIA:       JSR     PC,HMS
                                ;                               ADR             ;ADDRESS OF THE TIME
                                ;                               ADRX            ;ADDRESS OF EXTENDED TIME BITS
                                
013424  004437  012244  HMS:    JSR     R4,SAV04        ;SAVE REGS. 0-4
013430  016600  000012          MOV     10.(SP),R0      ;GET ADR OF OF ADR TIME
013434  010003                  MOV     R0,R3           ;COPY IT INTO REG. 3
013436  005723                  TST     (R3)+           ;GET ADR OF EXTENDED TIME BITS ADR
013440  017000  000000          MOV     @(R0),R0        ;PUT THE TIME IN REG. 0
013444  017303  000000          MOV     @(R3),R3        ;PUT EXTENDED TIME BITS INTO REG. 3
013450  005001                  CLR     R1              ;ZERO MINUTES COUNTER
013452  005002                  CLR     R2              ;ZERO HOURS COUNTER
                                
013454  022700  007020  1$:     CMP     #3600.,R0       ;LESS THAN 1 HOUR?
013460  101013                  BHI     3$              ;YES, GO LOOK FOR MORE HOURS
013462  162700  007020          SUB     #3600.,R0       ;NO, SUBTRACT 1 HOUR'S WORTH OF SECONDS
013466  005202                  INC     R2              ;COUNT AN HOUR
013470  000771                  BR      1$              ;CHECK FOR MORE HOURS
                                
013472  022700  000074  2$:     CMP     #60.,R0         ;LESS THAN 1 MINUTE?
013476  101014                  BHI     4$              ;YES, ALL DONE
013500  162700  000074          SUB     #60.,R0         ;NO, SUBTRACT 1 MINUTE'S WORTH OF SECONDS
013504  005201                  INC     R1              ;COUNT A MINUTE
013506  000771                  BR      2$              ;CHECK FOR MORE MINUTES
                                
013510  005703          3$:     TST     R3              ;ANY EXTENDED TIME BITS ?
013512  001767                  BEQ     2$              ;NO, GO LOOK FOR MINUTES
013514  062702  000021          ADD     #17.,R2         ;YES, ADD IN 17. HOURS
013520  062700  010360          ADD     #4336.,R0       ;ADD IN 1 HR 12 MIN 16 SEC
013524  005303                  DEC     R3              ;GET RID OF ONE TIME EXTENDED BIT
013526  000752                  BR      1$              ;CONTINUE
                                
013530  010246  4$:     MOV     R2,-(SP)        ;PUT THE HOURS ON THE STACK
013532  004537  011744          JSR     R5,BDCNV        ;CONVERT IT TO ASCII
013536  015507                  HOLD5                   ;AND PUT IT HERE
013540  113737  015511  001353  MOVB    HOLD5+2,HOURS   ;LOAD HOURS
013546  013737  015512  001354  MOV     HOLD5+3,HOURS+1 ;LOAD HOURS
013554  010146                  MOV     R1,-(SP)        ;PUT THE MINUTES ON THE STACK
013556  004537  011744          JSR     R5,BDCNV        ;CONVERT IT TO ASCII
013562  015507                  HOLD5                   ;AND PUT IT HERE
013564  013737  015512  001356  MOV     HOLD5+3,MINITS  ;LOAD MINUTES
013572  010046                  MOV     R0,-(SP)        ;PUT THE SECONDS ON THE STACK
013574  004537  011744          JSR     R5,BDCNV        ;CONVERT IT TO ASCII
013600  015507                  HOLD5                   ;AND PUT IT HERE
013602  013737  015512  001360  MOV     HOLD5+3,SEKNDS  ;LOAD SECONDS
                                
013610  004737  012256          JSR     PC,RST04        ;RESTORE REGS 0-4
013614  062716  000004          ADD     #4,(SP)         ;SET PROPER RETURN PC
013620  000207                  RTS     PC              ;AND RETURN
```

```
2334
2335
2336                                    ;       THIS IS THE CALLING SEQUENCE FOR THE CKHUNG ROUTINE
2337
2338  013622  013706  001136            HUNGCK: MOV     SPSAV,SP        ;RESTORE MONITOR STACK POINTER
2339  013626  005046                            CLR     -(SP)           ;CLEAR PSW ON STACK
2340  013630  012746  013636                    MOV     #1$,-(SP)       ;SETUP RETURN PC
2341  013634  000002                            RTI                     ;LOAD NEW PSW AND CONTINUE
2342  013636  005037  001374            1$:     CLR     CSTART          ;INDICATE NO MODULES DOING EOP
2343  013642  000137  004550                    JMP     ENDSVG          ;GO CHECK FOR OTHER MODULES RUNNING
```



```
2344                                    ;GET WRITE BUFFER ADDRESS SERVICE
2345
2346  013646  010146                    GWBUF,: MOV     R1,-(SP)        ;SAVE REGISTER
2347  013650  010246                            MOV     R2,-(SP)
2348  013652  017601  000004                    MOV     @4(SP),R1       ;GET ADDRESS OF MODULE
2349  013656  010102                            MOV     R1,R2
2350  013660  062701  000074                    ADD     #WBUFPA,R1
2351  013664  013721  001212                    MOV     WBUF,(R1)+      ;LOAD ADDRESS OF WRITE BUFFER
2352  013670  013721  001214                    MOV     WBFEA,(R1)+     ;LOAD EXTENDED ADDRESS BITS
2353  013674  012111                            MOV     (R1)+,(R1)      ;MOVE SIZE REQUESTED INTO SIZE ALLOWED
2354  013676  021137  001236                    CMP     (R1),MXWBF      ;REQUESTED SIZE OK?
2355  013702  101402                            BLOS    1$              ;YES- BRANCH
2356  013704  013711  001236                    MOV     MXWBF,(R1)      ;NO- SET ALLOWED SIZE
2357  013710  105737  001021            1$:     TSTB    MDXCTR          ;IF NOT YET SETUP, SKIP
2358  013714  100410                            BMI     2$
2359  013716  126237  000005  001023            CMPB    XFLAG(2),WBFLG  ;IF SAME, HAS ALREADY USED THIS BANK
2360  013724  001404                            BEQ     2$
2361  013726  105162  000005                    COMB    XFLAG(2)        ;IF NOT, SET FLAG TO SAME
2362  013732  105337  001021                    DECB    MDXCTR          ;AND COUNT DOWN MODULES
2363  013736  012602                    2$:     MOV     (SP)+,R2        ;RESTORE REGISTERS
2364  013740  012601                            MOV     (SP)+,R1
2365  013742  062716  000002                    ADD     #2,(SP)         ;SETUP RETURN PC
2366  013746  105337  001035                    DECB    ROT
2367  013752  100005                            BPL     3$
2368  013754  113737  001305  001035            MOVB    ROTN,ROT
2369  013762  004737  013000                    JSR     PC,ROTBUF       ;GO CHANGE WRITE BUFFER ADDRESS
2370  013766  000002                    3$:     RTI                     ;RETURN
2371
2372
2373
2374  013770  010146                    GETPA,: MOV     R1,-(SP)        ;SAVE REG, 1
2375  013772  017601  000002                    MOV     @2(SP),R1       ;GET ADDRESS OF TABLE
2376  013776  012121                            MOV     (R1)+,(R1)+     ;COPY VA INTO PA
2377  014000  005011                            CLR     (R1)            ;CLEAR EA BITS
2378  014002  012601                            MOV     (SP)+,R1        ;RESTORE REG, 1
2379  014004  062716  000002                    ADD     #2,(SP)         ;SETUP RETURN PC
2380  014010  000002                            RTI                     ;RETURN
2381
2382
```

```
                                        ;THIS ROUTINE ALLOWS RH70 DEVICES TO BE RUN WITH THE STANDARD MONITOR.
                                        ;
                                        ;
014012  010146                  MAP22,: MOV     R1,-(SP)        ;SAVE REG. 1
014014  017601  000002                  MOV     @2(SP),R1       ;GET THE 1ST ADDRESS OF THE TABLE
014020  012161  000002                  MOV     (R1)+,2(R1)     ;GIVE BACK THE SAME 18 BITS
014024  011161  000004                  MOV     (R1),4(R1)      ;THAT WERE GIVEN TO BE CONVERTED
014030  006261  000004                  ASR     4(R1)           ;SHIFT BACK FROM POSITIONS 4 AND 5
014034  006261  000004                  ASR     4(R1)           ;TO POSITIONS 0 AND 1
014040  006261  000004                  ASR     4(R1)           ;
014044  006261  000004                  ASR     4(R1)           ;
014050  012601                          MOV     (SP)+,R1        ;RESTORE REG. 1
014052  062716  000002                  ADD     #2,(SP)         ;SKIP OVER VA TO GET THE RETURN PC
014056  000002                          RTI                     ;RETURN
014060  000207                  MVCODE: RTS     PC              ;DO NOTHING
```



```
                                        ;PURE MESSAGES
                                        ;
014062  042045  041505  054057  TITLE:  .ASCIZ  '%DEC/X11, DXQAE-D SHORT MONITOR'
014070  030461  020056  054104
014076  040521  026505  020104
014104  044123  051117  020124
014112  047515  044516  047524
014120  000122
014122  027045     000          DOT:    .ASCIZ  '%.'
014125     045     000          CR:     .ASCIZ  '%'
014127     045  000045          CRCR:   .ASCIZ  '%%'
014132  040440  000124          AAT:    .ASCIZ  ' AT'
014136  051045  047125  051440  SUMARY: .ASCIZ  '%RUN SUMMARY'
014144  046525  040515  054522
014152     000
                014155          ACRLF=.+2
014153     136  022503     000  CTRLC:  .ASCIZ  '^C%'
014157     045  020045  052521  QOVRFL: .ASCIZ  '%% QUEUE OVERFLOW --- SYSTEM CRASH%'
014164  052505  020105  053117
014172  051105  046106  053517
014200  026440  026455  051440
014206  051531  042524  020115
014214  051103  051501  022510
014222     000
014223     045  020045  047111  INVTRP: .ASCIZ  '%% INVALID TRAP CALL --- SYSTEM CRASH%'
014230  040526  044514  020104
014236  051124  050101  041440
014244  046101  020114  026455
014252  020055  054523  052123
014260  046505  041440  040522
014266  044123  000045
014272  041045  042101  041440  INVCMD: .ASCIZ  '%BAD CMD%'
014300  042115  000045
014304  041445  052116  020114  INVCD1: .ASCIZ  '%CNTL C ONLY%'
014312  020103  047117  054514
014320  000045
014322  047045  020117  052113  INVCD2: .ASCIZ  '%NO KT11%'
014330  030461  000045
014334  047045  020117  052113  INVCD3: .ASCIZ  '%NO KT11%'
014342  030461  000045
014346  047045  020117  047515  INVCD4: .ASCIZ  '%NO MODS SEL%'
014354  051504  051440  046105
014362  000045
014364  041045  042101  040440  INVADR: .ASCIZ  '%BAD ADR/DATA%'
014372  051104  042057  052101
014400  022501     000
014403     045  040502  020104  INVNAM: .ASCIZ  '%BAD NAME%'
014410  040516  042515  000045
014416  045445  052502  020106  KBOFLO: .ASCIZ  '%KBUF OFLO'
014424  043117  047514     000
014431     045  052045  020117  CLR40:  .ASCIZ  '%%TO EXERCISE LOAD MEDIUM YOU MUST CLEAR LOC 40%'
014436  054105  051105  044503
014444  042523  046040  040517
014452  020104  042515  044504
014460  046525  054440  052517
014466  046440  051525  020124
```

```
2454  014474  046103  040505  020122
2455  014502  047514  020103  030064
2456  014510  000045
2457  014512  051445  051531  042440  SYSERR: .ASCIZ  '%SYS ERR (VIRT,PC,PSW,SP,VECT): '
2458  014520  051122  024040  044526
2459  014526  052122  050056  026103
2460  014534  051520  026127  050123
2461  014542  053054  041505  024524
2462  014550  020072  000040
2463  014554  022445  047520  042527  PWRFAI: .ASCIZ  '%%POWER FAILURE%%'
2464  014562  020122  040506  046111
2465  014570  051125  022505  000045
2466  014576  051045  047125  046440  MEMCHG: .ASCIZ  '%RUN MEMORY TESTS'
2467  014604  046505  051117  020131
2468  014612  042524  052123  000123
2469  014620  044045  046101  022524  HLT:    .ASCIZ  '%HALT%%'
2470  014626  000045
```



```
2471  014630  053445  044522  042524  ROTENB: .ASCIZ  '%WRITE BUFFER ROTATION ENABLED, RANGE: '
2472  014636  041040  043125  042506
2473  014644  020122  047522  040524
2474  014652  044524  047117  042440
2475  014660  040516  046102  042105
2476  014666  020056  040522  043516
2477  014674  035105  000040
2478
2479  014700  053445  044522  042524  ROTDIS: .ASCIZ  '%WRITE BUFFER ROTATION DISABLED'
2480  014706  041040  043125  042506
2481  014714  020122  047522  040524
2482  014722  044524  047117  042040
2483  014730  051511  041101  042514
2484  014736  000104
2485  014740  045445  030524  020061  KTONM:  .ASCIZ  '%KT11 ENABLED'
2486  014746  047105  041101  042514
2487  014754  000104
2488  014756  051445  044527  041524  NOSWR:  .ASCIZ  '%SWITCH REGISTER AT LOC. 1004%'
2489  014764  020110  042522  044507
2490  014772  052123  051105  040440
2491  015000  020124  047514  027103
2492  015006  030440  030060  022464
2493  015014     000
2494          015016                          .EVEN
```

```
                                ;ROUTINE TO LOAD WORST CASE UNIBUS PATTERN IN ALL FREE MEMORY

015016  004437  012244          LOADWC: JSR     R4,SAV04
015022  013702  001204                  MOV     LCOR12,R2       ;SETUP 12 BIT STARTING ADDRESS
015026  063702  001222                  ADD     OFFSET,R2
015032  023702  001206                  CMP     HCOR12,R2       ;ANY MEMORY TO LOAD ?
015036  001425                          BEQ     8$              ;NO, GET OUT
015040  013700  001200                  MOV     LOCORE,R0       ;IF NO KT11, R0 WILL CONTAIN PHYSICAL ADDRESS
015044  012704  015132          2$:     MOV     #WCASE,R4       ;LOAD ADDRESS OF WORST CASE DATA TABLE
015050  105737  001252          3$:     TSTB    KTPRES          ;USE KT11?
015054  001404                          BEQ     5$              ;NO- BRANCH
015056  012700  040000                  MOV     #40000,R0       ;SET VIRTUAL ADDRESS TO THE START OF PAGE 2
015062  010237  172344                  MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL ADDRESS
015066  012703  000400          5$:     MOV     #256.,R3        ;SET UP COUNTER
015072  011420                  6$:     MOV     (R4),(R0)+      ;LOAD DATA
015074  005303                          DEC     R3
015076  001375                          BNE     6$
015100  062702  000010                  ADD     #10,R2          ;AFTER EACH 256 WORDS, REMAP
015104  020237  001206                  CMP     R2,HCOR12       ;DONE?
015110  002403                          BLT     7$              ;NO- BRANCH
015112  004737  012256                  JSR     PC,RST04
015116  000207                          RTS     PC              ;YES- EXIT
015120  005724                  7$:     TST     (R4)+           ;MOVE DATA POINTER
015122  020427  015232                  CMP     R4,#WCASEE      ;TO NEXT WORST CASE VALUE
015126  103750                          BLO     3$              ;AND CONTINUE
015130  000745                          BR      2$

015132  177776  000001  177775  WCASE:  .WORD   177776,1,177775,2,177773,4,177767,10
015140  000002  177773  000004
015146  177767  000010
015152  177757  000020  177737          .WORD   177757,20,177737,40,177677,100,177577,200
015160  000040  177677  000100
015166  177577  000200
015172  177377  000400  176777          .WORD   177377,400,176777,1000,175777,2000,173777,4000
015200  001000  175777  002000
015206  173777  004000
015212  167777  010000  157777          .WORD   167777,10000,157777,20000,137777,40000,77777,100000
015220  020000  137777  040000
015226  077777  100000
015232                          WCASEE:
```



```
                                ;COMMAND TABLE. KEYBOARD COMMANDS ARE MATCHED AGAINST IT.
                                ;THE MATCH MUST BE EXACT. NO ABBREVIATIONS ARE ALLOWED.

015232                          COMTAB:
015232                                  TOKN    ""RUN",RUN
015232  052522  000116                  .ASCIZ  %RUN%
015240  007420                          .WORD   RUN
015242                                  TOKN    ""MOD",MOD
015242  047515  000104                  .ASCIZ  %MOD%
015250  010574                          .WORD   MOD
015252                                  TOKN    ""SEL",SEL
015252  042523  000114                  .ASCIZ  %SEL%
015260  010422                          .WORD   SEL
015262                                  TOKN    ""DES",DES
015262  042504  000123                  .ASCIZ  %DES%
015270  010440                          .WORD   DES
015272                                  TOKN    ""MAP",MAP
015272  040515  000120                  .ASCIZ  %MAP%
015300  010254                          .WORD   MAP
015302                                  TOKN    ""FILL",FILL
015302  044506  046114     000          .ASCIZ  %FILL%
015310  010566                          .WORD   FILL
015312                                  TOKN    ""KTOFF",KTOFF
015312  052113  043117  000106          .ASCIZ  %KTOFF%
015322  010522                          .WORD   KTOFF
015324                                  TOKN    ""KTON",KTON
015324  052113  047117     000          .ASCIZ  %KTON%
015332  010522                          .WORD   KTON
015334                                  TOKN    ""PON",PON
015334  047520  000116                  .ASCIZ  %PON%
015342  010522                          .WORD   PON
015344                                  TOKN    ""POFF",POFF
015344  047520  043106     000          .ASCIZ  %POFF%
015352  010522                          .WORD   POFF
015354                                  TOKN    ""ROTON",ROTON
015354  047522  047524  000116          .ASCIZ  %ROTON%
015364  010526                          .WORD   ROTON
015366                                  TOKN    ""ROTOFF",ROTOFF
015366  047522  047524  043106          .ASCIZ  %ROTOFF%
015374     000
015376  010542                          .WORD   ROTOFF
015400     015     000                  .BYTE   15,0            ;CR
015402     004     000                  .BYTE   4,0
015404  007364                          .WORD   COMCD2
015406  000000  000000  000000          .WORD   0,0,0
```

```
 2580                                   ;IMPURE MESSAGES.
 2581
 2582  015414  020045  020040  020040  AMODNM: .ASCII  '%      AT '
 2583  015422  040440  020124
 2584  015426  020040  020040  020040  APC:    .ASCII  '       STAT '
 2585  015434  051440  040524  020124
 2586  015442  020040  020040  020040  AMDSTA: .ASCII  '      '
 2587  015450  020040  040520  041523  MDIRE:  .ASCII  '  PASCNT '
 2588  015456  052116     040
 2589  015461     040  020040  020040  APSCNT: .ASCII  '     . ERRCNT '
 2590  015466  020056  042440  051122
 2591  015474  047103  020124
 2592  015500  020040  020040  027040  AERRS:  .ASCIZ  '     .'
 2593  015506     000
 2594                                   .ODD
 2595  015507     040  020040  020040  HOLD5:  .ASCIZ  '     '
 2596  015514     000
 2597  015515     040  020040  020040  HOLD6:  .ASCIZ  '        '
 2598  015522  020040  000040
 2599  015526  020040  020040  020040  HOLD8:  .ASCIZ  '          '
 2600  015534  020040  020040     000
 2601  015541     040  020040  020040  AEND:   .ASCII  '       PC='
 2602  015546  020040  041520     055
 2603  015553     040  020040  020040  AEND1:  .ASCII  '       APC='
 2604  015560  020040  050101  026503
 2605  015566  020040  020040  020040  AEND2:  .ASCIZ  '       '
 2606  015574  000040
 2607  015576  020040  040520  051523  $PASS:  .ASCII  '  PASS# '
 2608  015604  020043
 2609  015606  020040  020040  027040  APASS:  .ASCIZ  '     .'
 2610  015614     000
 2611  015615     040  051104  050117  MODEND: .ASCIZ  ' DROPPED%'
 2612  015622  042520  022504     000
 2613  015627     105  042116  040520  APSEND: .ASCII  'ENDPAS='
 2614  015634  026523
 2615  015636  020040  020040  027040  BPSCNT: .ASCIZ  '     .'
 2616  015644     000
 2617  015645     045  051503  040522  AERROR: .ASCII  '%CSRA '
 2618  015652     040
 2619  015653     040  020040  020040  ACSRAC: .ASCII  '       CSRC '
 2620  015660  020040  051503  041522
 2621  015666     040
 2622  015667     040  020040  020040  ACSRC:  .ASCII  '       STATC '
 2623  015674  020040  052123  052101
 2624  015702  020103
 2625  015704  020040  020040  020040  ASTATC: .ASCIZ  '       '
 2626  015712  000040
 2627  015714  041445  051123  020101  ADTERR: .ASCII  '%CSRA '
 2628  015722  020040  020040  020040  ADTE6:  .ASCII  '       S/B '
 2629  015730  051440  041057     040
 2630  015735     040  020040  020040  ADTE3:  .ASCII  '       WAS '
 2631  015742  020040  040527  020123
 2632  015750  020040  020040  020040  ADTE2:  .ASCII  '       WRADR '
 2633  015756  053440  040522  051104
 2634  015764     040
 2635  015765     040  020040  020040  ADTE5:  .ASCII  '       RDADR '
```



```
 2636  015772  020040  042122  042101
 2637  016000  020122
 2638  016002  020040  020040  020040  ADTE4:  .ASCIZ  '       '
 2639  016010  000040
 2640  016012  042040  052101  020101  ADTE4A: .ASCIZ  ' DATA ERROR'
 2641  016020  051105  047522  000122
 2642  016026  047527  042122  020043  ADTEX:  .ASCII  'WORD# '
 2643  016034  020040  020040  027040  ADTE7:  .ASCIZ  '     .%'
 2644  016042  000045
 2645  016044  042440  051122  020043  ERRNMB: .ASCII  ' ERR# '
 2646  016052  020040  020040  027040  AERNMB: .ASCIZ  '     .'
 2647  016060     000
 2648  016061     045  042515  020115  MERR1:  .ASCII  '%MEM FAILURE- PA '
 2649  016066  040505  046111  051125
 2650  016074  026505  050040  020101
 2651  016102  020040  020040  020040  MERR2:  .ASCII  '       S/B '
 2652  016110  051440  041057     040
 2653  016115     040  020040  020040  MERR3:  .ASCII  '       WAS '
 2654  016122  020040  040527  020123
 2655  016130  020040  020040  020040  MERR4:  .ASCIZ  '       '
 2656  016136  000040
 2657  016140  020045  020040  020040  AERNM:  .ASCII  '%       HAD '
 2658  016146  044040  042101     040
 2659  016153     040  020040  020040  AERCT:  .ASCII  '     . DATA ERRORS OUT OF '
 2660  016160  020056  040504  040524
 2661  016166  042440  051122  051117
 2662  016174  020123  052517  020124
 2663  016202  043117     040
 2664  016205     040  020040  020040  AERTOT: .ASCIZ  '     . WORDS READ%'
 2665  016212  020056  047527  042122
 2666  016220  020123  042522  042101
 2667  016226  000045
 2668  016230     040     040     040  DECVAL: .BYTE   40,40,40,40,40,40
 2669  016233     040     040     040
```

```
                                ;BUFFER AREAS.
                                
                                        .EVEN
016236                          MODQ:                   ;ROOM FOR 20 MODULE POINTERS
016306                          KBUF:                   ;KEYBOARD BUFFER, KBUFL LONG
016346  000100                          .BLKW   64.     ;MONITOR STACK AREA
016546                          SPBOT:
016546  000020                          .BLKW   16.     ;SPECIAL MONITOR STACK AREA
016606                          SPSPEC:
016606  000310                  IOQ:    .BLKB   IOQL
017116  000454                  TYPEQ:  .BLKB   TYPQL
                                
017572                                  .=.             ;FIRST ADDRESS AFTER QUEUES
014000                          BUFSIZ=AMODNM-START
                                
```



```
                                ;ROUTINE TO DETERMINE WHETHER WRITE BUFFER ROTATION SHOULD TAKE PLACE,
                                ;AND TO DETERMINE CORE LIMITS OF BUFFER ROTATION. ALSO TO DETERMINE USE OF
                                ;RTI OR RTT INSTRUCTION.
016606                                  .=IOQ
016606  012737  016624  000010  SETBUF: MOV     #1$,RESIV       ;SET UP A RESERVED INSTRUCTION TRAP.
016614  005046                          CLR     -(SP)           ;CLEAR A WORD ON THE STACK FOR A NEW PS
016616  012746  016654                  MOV     #2$,-(SP)       ;SET UP TO EXIT WITH RTT INSTRUCTION.
016622  000006                          RTT                     ;IF RTT NOT VALID IT WILL TRAP OUT.
016624  012737  000002  005000  1$:     MOV     #PTI,TRCIA      ;TRAP COMES HERE, CHANGE RTT'S TO RTI'S.
016632  012737  000002  002620          MOV     #RTI,RTT1       ;  BY SETTING THE INSTRUCTION VALUE
016640  013737  001336  001346          MOV     CHKL2,CHKL1     ;CHANGE SET OF CHKSUM WORDS BECAUSE
016646  013737  001340  001350          MOV     CHKH2,CHKH1     ;OF RTI'S REPLACING RTT'S
                                                                ;OF ALL RTT'S TO THOSE OF RTI'S
016654  012737  012524  000010  2$:     MOV     #RESINT,RESIV   ;RESTORE THE RESERVED INSTRUCTION TRAP VECTOR
016662  105037  001301                  CLRB    MDL45           ;NOT A 45
016666  105037  001252                  CLRB    KTPRES          ;NO KT11
016672  105037  001261          STBFA:  CLRB    ROTI            ;ASSUME NO BUFFER ROTATION.
016676  105037  001263                  CLRB    RELOC           ;ASSUME NO CODE RELOCATION
016702  013737  000000  001200          MOV     0,LOCORE        ;GET LOCORE ADDR, FROM LOC 0
016710  001004                          BNE     1$              ;BR IF LINKED WITH CONFIG/LINKER PROGRAM
                                                                ;IF NOT ASSUMES NOT IN CHAIN MODE
016712  005137  001200                  COM     LOCORE          ;INDICATE SETBUF HAS BEEN RUN
016716  000137  017214                  JMP     SETXT           ;GO PREPARE TO EXIT SETBUF ROUTINE
016722  004537  012174          1$:     JSR     R5,TOP12S       ;GET TOP 12 BITS OF LOCORE PHYS ADDR
016726  001200                          LOCORE                  ;AND PUT IT INTO LCOR12
016730  032737  000777  001200          BIT     #777,LOCORE     ;IS IT ON A 256 WORD BOUNDARY ?
016736  001414                          BEQ     2$              ;YES, CONTINUE
016740  042737  000777  001200          BIC     #777,LOCORE     ;NO, CLEAR OUT LOWER BITS
016746  042737  000007  001204          BIC     #7,LCOR12       ;ROUNDING OFF TO NEXT HIGHEST 256 BOUNDARY
016754  062737  001000  001200          ADD     #1000,LOCORE    ;ROUND UP TO THE NEXT 256 WORD BOUNDARY
016762  062737  000010  001204          ADD     #10,LCOR12      ;ALSO INDICATE THAT BOUNDARY IN LOCOR12
016770  013737  000042  001206  2$:     MOV     42,HCOR12       ;ARE WE IN CHAIN MODE?
016776  001403                          BEQ     KTTEST          ;IF NO,GO DETERMINE IF KT IS TO BE USED
017000  112737  177777  001260          MOVB    #-1,CHN         ;IF YES, SET THE CHAIN INDICATOR
017006  105737  001252          KTTEST: TSTB    KTPRES          ;IS THE KT11 AVAILABLE FOR USE?
017012  001035                          BNE     SETKT           ;IF YES, GO SIZE MEMORY USING IT
017014  012737  017056  000004          MOV     #5$,4           ;IF NO, ONLY SIZE MEMORY UP TO 28K
017022  012700  174000                  MOV     #-4000,R0       ;INITIALIZE LOOP COUNTER TO POINT TO BANK 0
017026  005001                          CLR     R1              ;INITIALIZE THE COUNTER OF 1K CHUNKS(BANKS)
017030  062700  004000          4$:     ADD     #4000,R0        ;POINT TO THE FIRST WORD IN THE NEXT 1K BANK
017034  062710  000000                  ADD     #0,(R0)         ;WILL TRAP IF LOC DOESN'T EXIST
017040  005201                          INC     R1              ;IF NO, CONTINUE PROCESING AT 4$
                                                                ;IF YES,UP THE MEMORY BANK COUNT
017042  022701  000034                  CMP     #28.,R1         ;HAVE 28 BANKS BEEN COUNTED?
017046  001370                          BNE     4$              ;IF NO, GO CONTINUE SIZING
017050  062700  004000                  ADD     #4000,R0        ;IF YES, INDICATE THE NUMBER OF WORDS IN R0
017054  000401                          BR      6$              ;SKIP OVER THE NEXT INSTRUCTION
017056  022626                  5$:     CMP     (SP)+,(SP)+             ;IF THERE WAS NO LOCATION AT LAST TRY,
                                                                ;REMOVE THE TRAP FROM THE STACK
017060  000300                  6$:     SWAB    R0              ;ISOLATE THE 12 HGH ORDER BITS, FIRST,
                                                                ;GET THE MOST SIGNIFICANT BITS IN THE LOW BYTE
017062  006300                          ASL     R0              ;ALIGN THE BITS TO ACCOUNT FOR
017064  006300                          ASL     R0              ;EXTENDED ADDRESS BITS (17 + 18)
017066  042700  170037                  BIC     #170037,R0      ;ELIMINATE THE UNWANTED BITS FROM R0
017072  105737  001260                  TSTB    CHN             ;ACT11 CHAIN?
017076  100003                          BPL     SETCOM          ;NO= DON'T ADD ROOM FOR MONITOR
```

```
 2741  017100  162700  000100                SUB     #100,R0         ;YES- ALLOW ROOM FOR MONITOR
 2742  017104  000400                      BR      SETCOM          ;GO TEST THE MEMORY
 2743  017106                        SETKT:
 2744  017106  012737  012516  000004  SETCOM: MOV     #BUSERR,4       ;RESET BUS ERROR TRAP.
 2745  017114  010037  001206                MOV     R0,HCOR12       ;HI ADDR TO HCOR12.
 2746  017120  004737  017222          SETCM1: JSR     PC,TSTMEM       ;TEST MEMORY FOR BASIC FUNCTIONING
 2747                                          ;VIA ADDRESS UP/DOWN TEST
 2748  017124  013700  001206                MOV     HCOR12,R0       ;DIFF BETWEEN LCOR12 AND HCOR12
 2749  017130  010037  001210                MOV     R0,HCORSV       ;INITIALIZE SAVE LOCATION FOR KTON/OFF COMMANDS
 2750  017134  163700  001204                SUB     LCOR12,R0       ;MUST BE AT LEAST 40
 2751  017140  022700  000040                CMP     #40,R0          ;IS IT?
 2752  017144  101023                        BHI     SETXT           ;FORGET IT IF NOT.
 2753  017146  105237  001261                INCB    ROTI            ;ALLOW ROTATION.
 2754  017152  105737  001252                TSTB    KTPRES          ;NO RELOCATION IF NO KT11
 2755  017156  001413                        BEQ     1$
 2756  017160  022700  000400                CMP     #400,R0         ;IF 8K OR GREATER, ALLOW CODE RELOCATION
 2757  017164  101010                        BHI     1$              ;BRANCH IF NOT
 2758  017166  013737  001206  001220        MOV     HCOR12,OFFMX    ;CALCULATE THE HIGHEST LOC TO WHICH
 2759  017174  163737  001204  001220        SUB     LCOR12,OFFMX    ;THE PROGRAM CAN BE RELOCATED
 2760  017202  105237  001263                INCB    RELOC           ;ENABLE RELOCATION
 2761  017206  004737  015016          1$:   JSR     PC,LOADWC       ;LOAD WORST CASE UNIBUS NOISE PATTERN
 2762  017212  000005                        RESET                   ;TURN OFF KT11 IF IN USE
 2763  017214                        SETXT:
 2764  017214  005037  000000                CLR     0               ;CLEAR LOC 0.
 2765  017220  000205                        RTS     R5              ;EXIT.
```



```
 2766                                   ;MEMORY ADDRESS UP/DOWN TEST
 2767
 2768  017222  023737  001206  001204  TSTMEM: CMP     HCOR12,LCOR12   ;ANY MEMORY TO TEST ?
 2769  017230  001001                        BNE     5$              ;YES, CONTINUE
 2770  017232  000207                        RTS     PC              ;NO, RETURN
 2771  017234  013702  001204          5$:   MOV     LCOR12,R2       ;LOAD 12 BIT LOCORE ADDRESS
 2772  017240  013701  001200                MOV     LOCORE,R1       ;R1 CONTAINS LOCORE PHYS ADDRESS
 2773  017244  012737  017626  000004        MOV     #NOMEM,4        ;LOAD TIME-OUT VECTOR
 2774  017252  105737  001252          1$:   TSTB    KTPRES          ;USE KT11?
 2775  017256  001002                        BNE     2$              ;YES- BRANCH
 2776  017260  010100                        MOV     R1,R0           ;NO- SET VA EQUAL TO PA
 2777  017262  000404                        BR      3$
 2778  017264  012700  040000          2$:   MOV     #40000,R0       ;SET VA TO PAGE 2
 2779  017270  010237  172344                MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL ADDRESS
 2780  017274  012703  000040          3$:   MOV     #32.,R3         ;COUNT 1 BLOCK
 2781  017300  010120                  4$:   MOV     R1,(R0)+        ;LOAD LOCATION WITH ITS ADDRESS
 2782  017302  062701  000002                ADD     #2,R1
 2783  017306  005303                        DEC     R3
 2784  017310  001373                        BNE     4$
 2785  017312  005202                        INC     R2              ;AFTER 1 BLOCK, UP PAGE ADDRESS FIELD
 2786  017314  020237  001206                CMP     R2,HCOR12       ;AT TOP?
 2787  017320  002754                        BLT     1$              ;NO- CONTINUE
 2788  017322  005302                        DEC     R2              ;YES- TEST GOING DOWN
 2789  017324  162701  000002                SUB     #2,R1
 2790  017330  105737  001252          TSTM1: TSTB    KTPRES          ;USE KT11?
 2791  017334  001002                        BNE     1$              ;YES- BRANCH
 2792  017336  010100                        MOV     R1,R0           ;NO- LOAD VIRTUAL ADDRESS POINTER WITH PHYS ADDRESS
 2793  017340  000404                        BR      2$
 2794  017342  012700  040076          1$:   MOV     #40076,R0       ;MAP VA TO TOP OF 1ST BLOCK IN PAGE 2
 2795  017346  010237  172344                MOV     R2,KIPAR2       ;MAP PAGE 2 TO PA
 2796  017352  012703  000040          2$:   MOV     #32.,R3         ;COUNT 1 BLOCK
 2797  017356  011037  001176          3$:   MOV     @R0,MEMWAS
 2798  017362  020137  001176                CMP     R1,MEMWAS       ;DOES LOCATION CONTAIN ITS PA?
 2799  017366  001120                        BNE     MERR            ;NO- BRANCH
 2800  017370  162700  000002          4$:   SUB     #2,R0
 2801  017374  162701  000002                SUB     #2,R1
 2802  017400  005303                        DEC     R3
 2803  017402  001365                        BNE     3$
 2804  017404  005302                        DEC     R2              ;AFTER CHECKING 1 BLOCK, ADDRESS NEXT
 2805  017406  020237  001204                CMP     R2,LCOR12       ;DONE?
 2806  017412  002346                        BGE     TSTM1           ;NO- CONTINUE CHECKING
 2807  017414  013702  001206                MOV     HCOR12,R2       ;YES- SET TO LOAD MEMORY
 2808  017420  010201                        MOV     R2,R1           ;FROM TOP DOWN
 2809  017422  005302                        DEC     R2
 2810  017424  006301                        ASL     R1
 2811  017426  006301                        ASL     R1
 2812  017430  006301                        ASL     R1
 2813  017432  006301                        ASL     R1
 2814  017434  006301                        ASL     R1
 2815  017436  005301                        DEC     R1
 2816  017440  006301                        ASL     R1
 2817  017442  105737  001252          TSTM2: TSTB    KTPRES          ;USE KT11?
 2818  017446  001002                        BNE     1$              ;YES- BRANCH
 2819  017450  010100                        MOV     R1,R0           ;NO- LOAD VIRTUAL POINTER WITH
 2820                                          ;PHYSICAL ADDRESS
 2821  017452  000404                        BR      2$
```

```
 2822  017454  012700  040076          18$:    MOV     #40076,R0       ;LOAD VIRTUAL WITH TOP ADDRESS OF
 2823                                                                    ;1ST BLOCK IN PAGE 2
 2824  017460  010237  172344                  MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL
 2825  017464  012703  000040          28$:    MOV     #32.,R3         ;COUNT A BLOCK
 2826  017470  010110                  38$:    MOV     R1,@R0          ;LOAD LOCATION WITH ITS PHYS ADDRESS
 2827  017472  005110                          COM     @R0             ;COMPLEMENT IT
 2828  017474  162701  000002                  SUB     #2,R1
 2829  017500  162700  000002                  SUB     #2,R0
 2830  017504  005303                          DEC     R3
 2831  017506  001370                          BNE     38
 2832  017510  005302                          DEC     R2              ;AFTER LOADING A BLOCK, SET UP FOR NEXT
 2833  017512  020237  001204                  CMP     R2,LCOR12       ;DONE?
 2834  017516  002351                          BGE     TSTM2           ;NO- BRANCH
 2835  017520  005202                          INC     R2              ;YES- SETUP TO CHECK IT
 2836  017522  062701  000002                  ADD     #2,R1
 2837  017526  010104                          MOV     R1,R4
 2838  017530  105737  001252          TSTM3:  TSTB    KTPRES          ;USE KT11?
 2839  017534  001002                          BNE     1$              ;YES- BRANCH
 2840  017536  010400                          MOV     R4,R0           ;SET VIRTUAL ADDRESS EQUAL TO PHYS ADDRESS
 2841  017540  000404                          BR      2$
 2842  017542  012700  040000          1$:     MOV     #40000,R0       ;SET VIRTUAL ADDRESS TO START OF PAGE 2
 2843  017546  010237  172344                  MOV     R2,KIPAR2       ;MAP PAGE 2 TO PHYSICAL ADDRESS
 2844  017552  012703  000040          2$:     MOV     #32.,R3         ;COUNT A BLOCK
 2845  017556  011037  001176          3$:     MOV     @R0,MEMWAS
 2846  017562  010401                          MOV     R4,R1
 2847  017564  005101                          COM     R1
 2848  017566  020137  001176                  CMP     R1,MEMWAS       ;DOES LOCATION CONTAIN COMPLEMENT OF ITS PHYS ADDRESS?
 2849  017572  001016                          BNE     MERR            ;NO- BRANCH
 2850  017574  005720                  4$:     TST     (R0)+
 2851  017576  062704  000002                  ADD     #2,R4
 2852  017602  005303                          DEC     R3
 2853  017604  001364                          BNE     3$
 2854  017606  005202                          INC     R2              ;CHANGE 12 BIT ADDRESS AFTER CHECKING A BLOCK
 2855  017610  020237  001206                  CMP     R2,HCOR12       ;DONE?
 2856  017614  002745                          BLT     TSTM3           ;NO- BRANCH
 2857  017616  012737  012516  000004          MOV     #BUSERR,4       ;RESET THE TIME-OUT TRAP VECTOR
 2858  017624  000207                          RTS     PC              ;YES- RETURN
 2859
 2860  017626  022626                  NOMEM:  CMP     (SP)+,(SP)+             ;POP THE STACK POINTER TWICE
 2861  017630  012737  012516  000004  MERR:   MOV     #BUSERR,4       ;RESET THE TIME-OUT TRAP VECTOR
 2862  017636  105237  001303                  INCB    MEMERF
 2863  017642  010037  001164                  MOV     R0,MEMVA        ;SAVE FAILING ADDRESS
 2864  017646  104413  001164                  GETPAS,MEMVA
 2865  017652  010137  001174                  MOV     R1,MEMSBE       ;SAVE EXPECTED CONTENTS
 2866  017656  013746  001166                  MOV     MEMPA,-(SP)     ;CONVERT PHYSICAL ADDRESS TO ASCII
 2867  017662  013746  001170                  MOV     MEMEA,-(SP)
 2868  017666  004537  011524                  JSR     R5,OACNVX
 2869  017672  016102                          MERR2
 2870  017674  013746  001174                  MOV     MEMSBE,-(SP)    ;CONVERT SHOULD BE TO ASCII
 2871  017700  004537  011512                  JSR     R5,OACNV
 2872  017704  016115                          MERR3
 2873  017706  013746  001176                  MOV     MEMWAS,-(SP)    ;CONVERT WAS VALUE TO ASCII
 2874  017712  004537  011512                  JSR     R5,OACNV
 2875  017716  016130                          MERR4
 2876  017720  004537  012174                  JSR     R5,TOP128       ;GET NEW HIGH CORE VALUE
 2877  017724  001166                          MEMPA
```



```
 2878  017726  013737  001172  001206          MOV     MEM12,HCOR12
 2879  017734  042737  000007  001206          BIC     #7,HCOR12
 2880  017742  162737  000040  001206          SUB     #40,HCOR12
 2881  017750  000207                          RTS     PC
```

```
 2682
 2683          000001                          .END
```



```
AAT     014132          2409#
ACRLF = 014155           874    889    892   1210   1224   2413#
ACSR  = 000052           168#
ACSRAC  015653           852   2619#
ACSRC   015667           846   2622#
ADDRHI  001044           265#  1758#
ADDRLO  001042           264#  1703   1757*
ADRERR  011062          1755   1762#  1780   1782
ADTERR  015714           865   2627#
ADTEX   016026           872   2642#
ADTE2   015750           784    820   2632#
ADTE3   015735           782    822   2630#
ADTE4   016002           773    828   2638#
ADTE4A  016012           864   2640#
ADTE5   015765           780    834   2635#
ADTE6   015722           762    840   2628#
ADTE7   016034           790   2643#
AEND    015541           696    857    919    943    991   2601#
AEND1   015553           917   2603#
AEND2   015566           915   2605#
AERCT   016153           729   2659#
AERNM   016140           726    735   2657#
AERNMB  016052           756    816   2646#
AERROR  015645           868   2617#
AERRS   015500          1640   2592#
AERTOT  016205           732   2664#
ALCLR   001332           386#
ALLBK   001322 G          37#   380#   511    952    962
AMDSTA  015442          1634   2586#
AMODNM  015414           333   1628   1644   1813   1817   1862   2582*  2683
APASS   015606           923   2609#
APC     015426          1631   2584#
APCLR   001334           387#
APSCNT  015461          1637   2589#
APSEND  015627           992   2613#
APTGAL= ****** U           1     27     28     30    207    307    407    432    434    467    492   1520   1526
                        2400   2406   2764
ASB   = 000056           172#  1047   1409*  1429*
ASTAT = 000054           170#
ASTATC  015704           844   2625#
ATE     001310           370#  1879*  1882*  1885*  1894   1907   1917   1928*
AWAS  = 000060           173#  1045   1062   1408*  1428*
BDCNV   011744 G          33#   728    731    755    789    815    922    987   1636   1639   1939#  2317   2322
                        2326
BIT0  = 000001           132#  1917   1928
BIT1  = 000002           131#
BIT10 = 002000           122#   530    540   1406   1539
BIT11 = 004000           121#   530    535   1535
BIT12 = 010000           120#   530    549    954    983   1545
BIT13 = 020000           119#   625    689    733    853    939   1004   1528   1531
BIT14 = 040000           118#   628    898   1529   1623   1665   1670
BIT15 = 100000           117#
BIT2  = 000004           130#
BIT3  = 000010           129#
BIT4  = 000020           128#
BIT5  = 000040           127#
```

```
BIT6  = 000100            126#   144
BIT7  = 000200            125#   1885
BIT8  = 000400            124#
BIT9  = 001000            123#   1002
BKCNT   001025            250#   960    1005*  1544*
BKPTR   001250            339#   442*   548*   596    1513*
BKQSVC  002622            502    647#
BKQUE   001012            239#
BKTIME  001304            365#   1015
BKTIMR  001034            259#   1012*  1015*
BPSCNT  015636            988    2615#
BRAKE   001013            240#   509    547*   949*   967    1001*  1272*  2150*  2172*
BREAK$= 104407            615    2105#
BREAK,  003136            617    715#
BUFSIZ= 014000            2683#
BUSERR  012516            189    2135#  2744   2857   2861
BUSEV   000004            189#   434    436*   441*
CADDR   001372            413    415#   687*   708*   721*   743*   804*   980*   1479*
CAST  = ****** G           36#   211
CCSRA   001144            298#   836*   848*   900*   903    906    907*
CDATA$= 104414            2115#
CDATA,  006626            1363#  2114
CDATA1  006652            1369#  1449
CDERR   001376            419#   1414
CDERR1  001400            420#   1369*  1370*  1413
CDFLG   001142            297#   1365*  1388   1448*
CDRET   007202            421    1444#
CHKH1   001350            400#   2696*
CHKH2   001340            396#   2696
CHKL1   001346            399#   2695*
CHKL2   001336            395#   2695
CHKSUM  001342 G           35#   397#
CHN     001260            347#   467    522    997    1245   1524   1601   2181   2191   2718*  2739
CHNOUT  012722            1252   2177   2181#
CKCONT  006632            1362   1365#
CLOCK   001321 G           34#   379#
CLOCKL  001324 G           33#   382#
CLOCKP  001326 G           33#   383#
CLRQUS  012272            446    461    1262   1268   1515   2045#
CLRVEC  012354            462    1514   2062#
CLR40   014431            458    2448#
CNT1    001307            369#
COMCON  007364            1460   1488#  1589   1618   1666   1722   2183
COMCO1  007360            1470   1487#
COMCO2  007364            1489#  2578
COMCO3  007374            1243   1492#  1676   1682   1690
COMCO4  007400            1493#  1499
COMQUE  001362            413#   712    908    1007   1483
COMTAB  015232            1463   2538#
CONVRT  011550            1880   1889#
CR      014125            993    1250   2407#
CRCR    014127            695    856    942    990    2408#
CSRA  = 000050            167#   906*   1080
CSTART  001374            416#   686*   688    709*   724*   746*   800*   805    905    981*   1257*  1261*  1263*
                          1480*  2342#
CTRLC   014153            1274   2414#
```



```
CTRLCA  006136            1168   1237#
CTRLCB  006142            973    1238#
CTRLCD  006224            1246   1252#
CTRLX   006230            1238   1254#  2146   2170
CXRET   001242            336#
DATCKS= 104417            2121#
DATCK,  006616            1361#  2120
DATERS= 104405            2101#
DATFLG  001315            375#   1361*  1363*  1436
DDPPTR  000042            206#
DECODE  007230            1225   1453#
DECVAL  016230            1941   2668#
DERRCT  001114            286#   1366*  1399*  1403*  1404   1411   1420   1428   1438
DERRXS= 104421            419    2125#
DERRX,  003244            674    741#
DERRX1  003312            751#   753
DES     010440            1655#  2550
DIRA    010274            1251   1617   1620#
DIRIND  001011            238#   1615*  1625   1641
DOT     014122            466    1492   1582   2159   2406#
DSTADR  001054            269#   556*   558*   620*   623    643    647*   716*   1422*  1435*
EMTV    000030            200#
ENDCOM  004314            693    749    806    909#   938    985
ENDPS$= 104402            2095#
ENDSVA  004446            901    939#
ENDSVB  004604            965    969#   1553
ENDSVC  004440            673    937#
ENDSVD  004622            961    966    968    972    974#   1003
ENDSVE  004474            947#   998    1006
ENDSVF  004610            959    971#
ENDSVG  004550            953    960#   2343
END$  = 104403            2097#
ERCNT = 000030            159#   754    807*   809*   814    896    1424*  1427*  1527*  1638
ERRIND  001017            244#   741*   795*   797*   799*   801    817    861    870    876
ERRLIM  001234            332#   896
ERRNMB  016044            859    2645#
ERRN$ = 104410            2107#
ERROR$= 104404            2099#
ERRPOS  001140            296#   764*   765*   768    776    785
ERRSVC  003524            673    799#
ERSVCA  003532            796    798    800#
ERSVCB  003600            808    810#
ERSVCC  003604            811#   813
ERSVC1  003516            673    797#
ERSVC2  003506            674    795#
ERSVND  004222            737    854    877    894#
ERSVX   004014            791    841    853#
EXIT$ = 104400            2091#
EXIT,   005030            1023#  1091   2090
EXIT1,  005032            1020   1024#  1096   1159
EXIT2,  005066            1035#  1051   1073   1101   1155
FAKESR  001004            232#   439    451
FILCNT  001254 G           36#   343#   1346   1719
FILCTR  001020            245#   1346*  1348*  1352*
FILL    010566            1696#  2556
FILLER  001255 G           36#   344#   1349
```

```
FILLID  001272            354#  1491*  1696*  1720
FILLNM  011206            725    918   1627   1798#
GETADR  011022           1701   1750#
GETNAM  011234           1661   1698   1809#
GETNUM  011070           1710   1750   1768#
GETPA$= 104413            824    830    909   2113#  2864
GETPA.  013770           2112   2374#
GTOK    007272           1463#
GTOKX   007276           1464#
GTOK1   007300           1465#  1473   1475
GTOK2   007322           1466   1474#
GWBUF$= 104412           2111#
GWBUF.  013646           2110   2346#
HARDSR= 177570             77#   435
HCORSV  001210            321#  2749#
HCOR12  001206 G           33#   319#  1596   1600   2228   2263   2500   2513   2716*  2745*  2748   2758   2768
                         2786   2807   2855   2878*  2879*  2880*
HLT     014620           1142   1144   2084   2469#
HMS     013424 G           34#  2288#
HNGFLG  001312            372#
HNGTIM  001330 G           35#   385#
HOLD5   015507           2318   2319   2320   2323   2324   2327   2328   2595#
HOLD6   015515            884    885   1729   1730   1744   1745   2597#
HOLD8   015526           1737   1738   2599#
HOP     011540           1883   1887#
HOURS   001353 G           34#   402#  2319*  2320*
HUNGCK  013622 G           35#  2338#
IE    = 000100            144#   475   1197   1248
INIT  = 000022            156#   556
INPUT   001650            468    472#  1493   1584
INVADR  014364           1762   2441#
INVCD1  014304           1459   2431#
INVCD2  014322           2434#
INVCD3  014334           2436#
INVCD4  014346           1588   2438#
INVCMD  014272           1487   2429#
INVNAM  014403           1851   2444#
INVTRP  014223           2083   2422#
IOBKID  001060            271#   490    515*   524*   531    542    564    566*   568    570*   584    591*   1558*
                         2149*  2171#
IOQ     016606            149    609   1142   1148   2050   2679#  2688
IOQL  = 000310            146#   149   2055   2679
IOQLIM= 017116            149#   607   1146
IOQSVA  002522            625#   653
IOQSVB  002604            641#
IOQSVC  002424            489    607#
IOQSVD  002570            635#  1442
IOQSVE  002514            563    622    624#   717
IOQSVN  002500            414    620#
IOQUE   001006            235#   487    610*  1016   1138   1150*  2045   2046
IOQ1    001074            277#  1140   1142*  1146   1148*  1149*  1151*  2051*
IOQ2    001076            278#   607    609*   611    612*  1140   2052*
IOTV    000020            195#
ITOA    011000           1706   1708   1742#  2152   2261
ITOAX   010710           1725#  2275
ITOA8   010744           1733#
```



```
ITOUT   011016           1731   1739   1746#
KBOFLO  014416           1229   2446#
KBPTR   001036            262#   469*   474*  1174   1193*  1194*  1204   1206*  1207   1211*  1216   1222*  1223*
                         1462   1481*  1768   1793*  1809   1857*  1869*
KBSRVC  005644            214   1164#
KBUF    016306            474   1174   1204   1211   1216   1223   2674#
KBUFL = 000040            145#  1174   2675
KIPAR0= 172340            101#
KIPAR1= 172342            102#
KIPAR2= 172344            103#  2507*  2779*  2795*  2824*  2843*
KIPAR3= 172346            104#
KIPAR7= 172356            105#
KIPDR0= 172300             96#
KIPDR1= 172302             97#
KIPDR2= 172304             98#
KIPDR3= 172306             99#
KIPDR7= 172316            100#
KTOFF   010522           1676*  2559
KTON    010522           1675*  2562
KTONM   014740           2485#
KTPRES  001252 G           37#   340#   463   2184   2504   2700*  2719   2754   2774   2790   2817   2838
KTTEST  017006           2717   2719#
KW11L = ****** G           34#   382
KW11P = ****** G           34#   383
KYBDV   000060            214#
LCLEAR= ****** G           35#   386
LCOR12  001204 G           33#   318#  1594   2498   2713*  2715*  2750   2759   2768   2771   2805   2833
LDBRK.  005460           1124#  2104
LDIOQ   005474           1125   1133#
LDTYPQ  005350           1041   1044   1046   1048   1055   1058   1060   1067   1077   1086   1094   1099   1103#
LOADWC  015016            465   2497#  2761
LOCK    001275            357#  1505*  2201   2232
LOCORE  001200            316#   443   1593   2260   2502   2703*  2706*  2709   2710   2712*  2714*  2772
LOGIC   012752            208   2188#
LOGICA  012762           2176   2190#
MAP     010254           1615#  2553
MAPREG= 170200             82#
MAPRET  001232            331#
MAP22$= 104415           2117#
MAP22.  014012           2116   2396#
MDIRE   015450           1616*  1643*  2587#
MDL45   001301            362#  2699*
MDXCNT  001022            247#   551*   576*   590    956*  2223
MDXCTR  001021            246#   448*   571*   590*  2213   2219   2223*  2357   2362*
MED   = 076600             83#
MEMCHG  014576           2255   2466#
MEMEA   001170            312#  2867
MEMERF  001303            364#  2252   2256*  2862*
MEMPA   001166            311#  2866   2877
MEM88E  001174            314#  2865*  2870
MEMVA   001164            310#  2863*  2864
MEMWAS  001176            315#  2797*  2798   2845*  2848   2873
MEM12   001172            313#  2878
MERR    017630           2799   2849   2861#
MERR1   016061           2254   2648#
MERR2   016102           2651#  2869
```

```
MERR3   016115          2653#  2872
MERR4   016130          2655#  2875
MFLAG   001311           371#
MINITS  001356           403#  2324#
MOD     010574          1697#  2544
MODADR  001122           289#   631*  1025   1042   1056   1070   1434
MODCNT  001015 G          37#   242#   957*   960    964    971   1541*  1549   1555   1572   1576
MODCTR  001016 G          37#   243#   520    562*   1555*
MODEND  015615           944   2611#
MODPTR  001040           263#   526*   527    548    596*   599*   649   1557*  1664   1704   1811*  1868*
MODQ    016236 G          33#   442    599    1513   1517   1557   1620   1667   1859   2673#
MORE    007314          1471#
MOVBNK  001230           330#
MOVFLG  001313           373#
MOVING  001273           355#
MSCONT  003010           682    685#
MSGD$ = 104422          1432   2127#
MSGD.   003150           674    721#
MSGD1.  003240           734    737#
MSGFLG  001314           374#   681*   683*   691
MSGN$ = 104411          2109#
MSGN.   003004           674    683#
MSGS$ = 104416          2119#
MSGS.   002774           674    681#
MSG$  = 104406           450    453    458    466    1111   1112   1143   1144   1196   1210   1224   1229   1249
                        1250   1274   1459   1487   1492   1582   1588   1644   1689   1730   1738   1745   1762
                        1851   2083   2084   2103#  2151   2159   2173   2254   2255   2259
MSG.    003110           674    707#
MSG1.   003126           690    702    711#
MVCODE  014060           460   1240   1586   2397#
MVDOWN  001276           358#
MXWBF   001236           333#  1600*  1603*  1604*  1605*  1606*  1607*  1608*  1609*  1611*  2354   2356
NEW   = ****** U          38    153    310    332    754    801    807    814    842    876    896   1010   1424
                        1527   1638   2093   2097   2105   2494   2589   2625   2645
NOMEM   017626          2773   2860#
NOSWR   014756           453   2488#
NSTOP   001320           378#   582*  1559*  2205   2234
NUMBEA  001052           268#   826    832    911
NUMBER  001046           266#   666*   748*   823*   824    829*   830    909    912
NUMBPA  001050           267#   825    831    910
OACNV   011512 G          33#   781    783    819    821    843    845    883    914   1630   1633   1743   1878#
                        2871   2874
OACNVX  011524           761    772    779    827    833    839    851    916   1728   1882#  2868
OACNV8  011532          1736   1885#
OASCEA  001112           284#  1878*  1887*  1909   1919   1924
OFFMX   001220           325#  2758*  2759*
OFFSET  001222 G          35#   326#   533    537    1533   1537   1551   1595   2226   2499
OFFSTD  001224           327#   459*  1239*  1506*  1585*
OFFSTM  001226           328#
OPEN  = 000000           116#   205    209    210    232    235    236    237    238    239    240    241    242
                         243    244    245    246    247    248    249    250    251    252    253    254    256
                         258    259    260    262    263    264    265    266    267    268    269    270    271
                         272    273    274    275    276    277    278    279    280    281    282    283    284
                         286    287    288    289    290    291    292    293    294    295    296    297    298
                         310    311    312    313    314    315    316    318    319    321    323    324    325
                         326    327    328    330    331    335    336    337    338    339    340    345    346
```



```
                         347    348    349    350    351    354    355    356    357    358    359    362    363
                         364    368    369    370    371    372    373    374    375    377    378    380    403
                         404    415    416    420
PASEND  004626           672    979#
PASREL  001033 G          37#   258#  1572*  1573*  1574*  1575*  1578*
PASTOT  001032           256#
PCLEAR= ****** G          35#   387
PDP70 = ****** U           1     27     28     30    353    446    460    760    824    909    916   1240   1516
                        1586   1592   2027   2141   2158   7215   2226   2383   2400   2406   2463   2498   2513
                        2576   2603   2619   2628   2635   2651   2748   2763   2864   2868
PIRQRG= 177772            91#
PIRQ$ = 000004           115#   426   1170
PIRQ.   005510           195   1137#
POFF    010522          1674#  2568
PON     010522          1673#  2565
POPSP = 005726           141#  1049   1061   1089   1095   1100   1169   1199   1209
POPSP2= 022626           142#
PPRES   001306           368#
PRTY4 = 000200           137#   217
PRTY5 = 000240           136#
PRTY6 = 000300           135#
PRTY7 = 000340           134#   190    192    196    199    215   1133   1254
PS    = 177776            75#
PSCNT = 000026           158#   559    920    982*   986   1526*  1635
PSENDA  004714           984    995    996#
PSENDB  004766          1000   1007#
PSW   = 177776            76#
PUSH  = 005746           139#
PUSH2 = 024646           140#
PWRCNT  001246 G          37#   338#  1508*  2168*
PWRDN   012636           198    433   2163#  2166
PWRFAI  014554          2173   2463#
PWRFV   000024           198#   433*  2164*  2166*
PWRUA   012712          2160   2174#
PWRUP   012646          2164   2166#
QOVRFL  014157          1111   1143   2415#
QTSTA   001760           495    497#
QTSTB   001772           493    498    500#
QTSTC   002004           491    501    503#
QUECNT  001300           360#   476    485
QUECTR  001027           252#   476*   480*   485*
QUETST  001702           480#   504    510    519    525    600    627    630    974   1038   1198   1320   1358
QUE$  = 104401          1225   2093#
QUE.    005470          1129#  2092
RAND$ = 104420          2123#
RAND.   012066          1977#  2122
RANNUM  001160 G          36#   308#  1979   1999*
RANWRK  001162           309#  1978   1998*
RBUFEA= 000070           177#
RBUFPA= 000066           176#
RBUFSZ= 000072           178#
RBUFVA= 000064           175#
RDCUA = 000103            86#
RDFGIN= 000104            87#
RDPBA = 000102            85#
RDSERV= 000101            84#
```

```
RDWHMI= 000022          88#
RELOC   001263 G        37#     350#    2230    2702#   2760#
RESIA   012530          2136    2138#
RESINT  012524          191     2137#   2698
RESIV   000010          191#    2689#   2698#
RETRY   001277          359#
RLOOP   012106          1981#   1997
RMODE   001257          346#    447#    503     1241    1244#   1259    1457    1512#   1522    1580    1583#   1587#   2175
ROT     001035          260#    2366#   2368#
ROTBUF  013000          2197#   2369
ROTB1   013170          2230#
ROTB2   013226          2229    2239#
ROTB3   013252          2231    2233    2244#
ROTCHK  013262          454     1681    2252#
ROTCNT  001264          351#    1273#   1688#   2203#   2208#
ROTCT   001031          254#    1562#   1564#   1565#   1566#   1567#   1570#   1571    2224
ROTDIS  014700          1689    2479#
ROTENB  014630          2259    2471#
ROTI    001261          348#    1560    2197    2257    2701#   2753#
ROTL    001262          349#    449#    1680#   1687#   2199    2211
ROTN    001305          366#    2368
ROTNUM  001266          352#    1273    1688    2208
ROTOFF  010542          1687#   2575
ROTON   010526          1680#   2571
ROTSIZ  001270          353#
ROTXT   013260          2198    2204    2210    2212    2214    2222    2238    2243    2245    2247#
RSTAT   001066          274#    624#    641     648#    1439#
RSTRT = 000062          174#    558     980
RST04   012256          1482    1931    1960    2000    2037#   2330    2515
RTTK$ = 104423          2129#
RTTK.   012056          1967#   2128
RTT1    002620          644#    2694#
RUN     007420          470     1505#   2541
RUNC    007664          1521    1549#
RUND    010102          1552    1585#
RUNRES  007476          970     1515#   2194
RUNRFL  001274          356#    554     1511#   2169#
RUNSVC  002012          509#
SAVHI   001120          288#    2262#   2272#   2274
SAVLO   001116          287#    2263#   2264#   2265#   2266#   2267#   2268#   2269#   2270#   2273
SAV04   012244          1367    1445    1461    1889    1939    1977    2029#   2288    2497
SBADR = 000052          169#    1373#
SFKNDS  001360          404#    2328#
SFL     010422          1649#   2547
SFTBUF  016606          445     2689#
SFTCM1  017120          2746#
SFTCOM  017106          2740    2742    2744#
SFTKT   017106          2720    2743#
SFTWBF  010126          1579    1592#   2244
SFTWB1  010166          1597    1600#   2242
SFTXT   017214          2707    2752    2763#
SHORT = 000000          1       28      146     222     286     307     340     379     405     446     455     549     855
                        940     969     977     983     989     996     998     999     1241    1250    1252    1269    1384
                        1395    1400    1408    1418    1504    1509    1511    1516    1592    1598    1613    1673    1691
                        2026    2062    2070    2137    2146    2148    2159    2163    2175    2178    2190    2227    2230
                        2247    2248    2262    2278    2335    2339    2344    2373    2397    2400    2429    2463    2471
```



```
                        2539    2576    2582    2617    2673    2675    2681    2699    2742    2743    2748    2882
$INPUT  007404          1497#   1709
$LIS    010454          1651    1660#
$LISA   010454          1661#
$LISB   010470          1509#   1649#   1655#   1665#
$LISC   010502          1662    1667#
$LISD   010506          1668#   1671
$LISE   010512          1510#   1650#   1656#   1670#
$LISL   010476          1663    1666#   1669
$MODE = 040000          94#
$PEOT   016546          432     472     1237    2145    2158    2167    2676#
$PCFLG  001010          237#    1453    1455#   1489#   1498#
$PCINT= 000024          157#    553
$P&AV   001136          295#    632#    1034    1050    1072    2338
$P&REC  016606          2140    2148    2678#
$R      001156 G        36#     307#    435#    437     439#    451     689     733     853     894     898     983     1406
$RETRN  001056          270#    1456    1497#
$SR0  = 177572          79#     2186#
$SR3  = 172516          80#
$START  001414          227     229     322     333     397     431#    2193    2683
$TAT  = 000020          154#    529     535     540     549     624     625     628     939#    947     954     999     1002
                        1004#   1439    1528#   1529    1531    1535    1539    1542    1545    1623    1632    1665#   1670#
$TAT1 = 000021          155#
$TBFA   016672          2701#
$TOP    001024          249#    513     518     950     958     2207#   2209    2236#
$UMARY  014136          1249-   2410#
$VR0  = 000032          160#    1026    1374
$VR1  = 000034          161#
$VR2  = 000036          162#
$VR3  = 000040          163#
$VR4  = 000042          164#
$VR5  = 000044          165#
$VR6  = 000046          166#    553#    633     1071#   1431#   1441
$VCMNT  001244          337#    1507#   2147#
$VSIRI  001256          345#    473#
$VSIRR  014512          2151    2457#
$VSITM  001316          376#
TABADR  001064          273#    685#    699     803#    878
TENWR   012044          1942    1963#
TFLG    001317          377#
TITLE   014062          450     2400#
TKB     001150 G        36#     304#    1165    1176
TKS     001146 G        36#     303#    475#    1164#   1197#   1248#   2046
TOPI2S  012174          2008#   2708    2876
TPB     001154 G        36#     306#    1356#
TPS     001152 G        36#     305#    1299#   1330#
TRAIX = 000024          2087#   2091#   2093#   2095#   2097#   2099#   2101#   2103#   2105#   2107#   2109#   2111#   2113#
                        2115#   2117#   2119#   2121#   2123#   2125#   2127#   2129#
TRCI    004772          193     1012#
TRCIA   005000          1014#   1017    2693#
TRCIB   005016          1018#
TRCIC   005002          1013    1015#
TRCIC   001070          275#    500     647     650#    1018#
TRCISW  001072          276#    648     1019#
TRCV    000014          193#
TRPINT  012404          202     2074#
```

```
FRPLIM  012516          2079    2129#
FRPLAB  012446          2078    2088#
FRP\    000034           202#
FRP%  = 011000           669    2078    2086#
FSTPEM  017222          2746    2768#
FSTP1   017330          2790#   2806
FSTP2   017442          2817#   2834
FSTP3   017530          2838#   2856
FTPIV   000064           216#
FTSFV1  005426           427     499    1290    1321#   1354    1355
FTYESY  001014           241#    494     660*    711*    902*    946*    996*
FTYEYT  001240           335#   1195*   1196
FTYINK  001410           427#   1290*   1338*   1344*   1350*   1354*
FTYIRV  001406           216     426#
FYPJDR  001104           281#   1292*   1321    1334*
FYPCTR  001110           283#    497*   1357#
FYPEAT  004376           722     727     730     744     747     751     811     927#
FYPE    006356           703     710     736     869     873     875     886     890     893    1282    1290*
FYPEB   006466          1322    1334#
FYPEC   006502          1337#
FYPED   006552          1350#
FYPEE   006546          1347    1349#   1353
FYPEL   006600          1336    1339    1345    1351    1356#
FYPEN   006340           694     855     863     941     989    1279#
FYPEQ   017116           148     659     930    1110    1116    2053    2054    2680#
FYPI1   006342          1280#   1283
FYPIIMS 017572           148#    657     928    1114
FYPCL = 000454           147#    148    2055    2680
FYPCUE  001007           236#    492     663*    933*   1106    1118*
FYPCX   005264          1080#
FYPC.   005324          1094*   2094    2096    2108    2118
FYPC1   001100           279#   1108    1110*   1114    1116*   1117*   1119*   2053*
FYPC1.  005336          1099#   2102
FYPC2   001102           280#    657     659*    661     662*    928     930*    931     932*   1108    2054*
FYPC2.  005250          1076#   2098    2100    2106
FYPC3.  005150          1054#   2124
FYPC4.  005102          1041#   2126
FYPIET  001106           282#   1291*   1331
FYPI1   001124           290#   1293*   1329
FYPI2   001126           291#   1294    1324
FYPI3   001130           292#
FYPI4   001132           293#
FYPI5   001134           294#
FYPCVC  002650           496     657#
FYP%AB  002732           669     672#
UIPJR0= 177640           111#
UIPJR1= 177642           112#
UIPJR2= 177644           113#
UIPJR7= 177656           114#
UIPCR0= 177600           107#
UIPCR1= 177602           108#
UIPCR2= 177604           109#
UIPCR7= 177616           110#
UMOIE = 140000            93#
WASJDR= 000054           171#   1410#
WBFIA   001214           323#   2241#   2352
```



```
WRFIG   001023           248#   1516*   2225*   2359
WBUF    001212           322#   1593*   2215    2239*   2351
WBUFEA= 000076           180#   1592*
WBUFPA= 000074           179#   1386    2350
WBUFRQ= 000100           181#
WBUFSZ= 000102           182#
WRUI12  001216           324#   1594*   1595*   1596    1604    2226    2227*   2228    2237*
WCAIE   015132 G          34#   2503    2522#
WCAIEE  015232 G          34#    398    2518    2534#
WTWIMT= 000222            89#
XCN%    001026           251#   1547*   1562    1568
XFLJG = 000005           153#    552*   2359    2361#
XFRILG  001030           253#
YES     001062           272#   1712    1752    1771*   1791*   1810*   1833    1835    1838*   1841    1848    1856*
ZERO    001202           317#
$GWIFC  001302           363#   1571*   2221*   2224*
$PAIS   015576           697     858    2607#
.       = 017752            31#    187#    200     204#    213#    220     226#    228#    381#    401#    402#   1136    1394
                          1396   1426    1468    1504    1543    1546    1610    2413    2494#   2541#   2544#   2547#   2550#
                          2553#  2556    2559#   2562    2565#   2568    2571#   2575    2675#   2677#   2679#   2680#   2682#
                          2688#

. ABS.  017752      000
        000000      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DX>AED,DXQAED/SOL/CRF/EQ:SHORT=DXQABM
 RUN-TIME:  5  8  1 SECONDS
 RUN-TIME RATIO:  191/14=13.1
 CORE USED:  10K  (20 PAGES)
```