B. E. M. P860

```
IDENT LOCAT
ENTRY     STB
EXTRN     I TC
EXTRN     I PR
EXTRN     I PP
EXTRN     I DISK
EXTRN     I LP
EXTRN     I CR
EXTRN     I CP
EXTRN     I PFAR
EXTRN     I LKM
EXTRN     I RTC
EXTRN     I ITCP
EXTRN     I MEMP
EXTRN     I ASR
EXTRN     I MHDL
EXTRN     INTAB
EXTRN     CVT
DATA      I PFAR   40
DATA      I LKM    42
DATA      I RTC    44
DATA      I MEMP   46
DATA      I ITCP   48
DATA      I TC     4A
DATA      I PR     4C
DATA      I PP     4E
DATA      I ASR    50
DATA      I DISK   52
DATA      I LP     54
DATA      I CR     56
DATA      I CP     58
DATA      I MHDL   5A
RES       18       5C        INTERRUPT LOCATIONS 14-31
DATA      INTAB              MASKABLE INT TABLE ADDRESS
DATA      CVT                CVT ADDRESS
RES       30                 MULTIPLEX AREA
RES       16                 INTERRUPT LOCATIONS 32-47
RES       16                 OVERFLOW AREA
DATA      /FFFF              OVERFLOW LOCATION
RES       61                 STACK AREA
STB DATA  /FFFF              STACK BASE
          END
 EOS
```

```
            IDENT    I LKM
*LKM INTERRUPT HANDLING,SEARCH IN LKMAL IF THE REQUESTED MODULE
*IS AVAILABLE.     IF THE MODULE IS NOT HERE,RTN WITH A1=-1
*                  IF THE MODULE IS HERE BRANCH ON IT
            ENTRY    I LKM
            EXTRN    LKMAL
EXTRN       DISPAT
            EXTRN    PTC61
            EXTRN    SYSAB
            EXTRN    M A00
STATUS      EQU      0
LKM         EQU      X'2804'              LKM SYNTAX
I LKM       MSR      8,A15
            RIT      /1D
            CWK      A15,/100
            RF(5)    LKM3                 STACK OVERFLOW
            LD       A2,20,A15            ADDRESS OF USER DATA
            LDR*     A1,A2                USER DATA
            LD       A3,-2,A2             USER LKM
            CWK      A3,LKM               TEST IF LKM
            RF(0)    LKM1                 SYNTAX O.K.
            AB.L     M A00                SYNTAX N.O.K. GO TO SIMUL
LKM3        LDK      A1,1                 OR STACK OFLOW
            HLT
LKM1        LDK      A3,2
            LDK      A6,0
            ADK      A1,0
            RF(1)    LKM4
            C1R      A1,A1
            ADK      A1,1
            LD       A6,PCT61+STATUS
            ANK      A6,/7F
            CWK      A6,/7F
            RF(4)    *+4
            HLT
            IM       PCT61+STATUS
            LD       A6,2,A2              USER INTADDR
            LDK      A3,4
LKM4        AD.S     A3,20,A15
*                                         IF PROCESSOR NOT HERE ,RTN
            LD       A2,LKMAL
            CWR      A1,A2                YF LKM GRAETER THAN
            RF(1)    LKM2                 TABLE LENGTH ,RTN
            SLL1     A1
            LD       A2,LKMAL,A1
            RF(0)    LKM2
            ABR      A2
LKM2        LDK.L    A7,-1
            AB.L     DISPAT
            END
             EOS
```

```
          IDENT    LKMAL
*THIS IS THE LIST OF LKM ADDRESSES
*THE FIRST LOCATION OF TABLE IS THE TABLE LENGTH
          ENTRY    LKMAL
          EXTRN    M IORM
          EXTRN    WAIT,EXIT
          EXTRN    GETBUF,FRBUFF
          EXTRN    PSMAC
          EXTRN    ABADR
LKMAL     DATA     7
          DATA     M IORM    1
          DATA     WAIT      2
          DATA EXIT          3
          DATA     GETBUF    4
          DATA     FRBUFF    5
          DATA     PSMAC     6
          DATA ABADR         7
          END
           EOS
```

```
            IDENT   CVT
            ENTRY   CVT
            ENTRY   CVTMSZ          MEMORY SIZE
            ENTRY   CVTSTB          STACK A15 BASE
            ENTRY   CVTSBA          SMALLEST BUFF AREA ADDRESS
            ENTRY   CVTBBA          BIGGEST BUFF AREA ADDRESS
            ENTRY   CVTBKA          BACKGROUND ADDRESS
            EXTRN   STB
CVT         EQU     *
CVTMSZ      DATA    /6000           SIZE = 12 K WORDS
CVTSTB      DATA    STB
CVTSBA      DATA    /0
CVTBBA      DATA    0
CVTBKA      DATA    /1800
            END
X ⟶          EOS
```

```
                IDENT   CPT
                ENTRY   CPT
CPT             DATA    4               LENTH
                DATA    /0004           00= PAGE NUMBER 04= NUMBER OF PAGES
                DATA    /F007           MASK1 STARTING FROM RIGHT PAGE=15....01.0
                DATA    /FFFF           MASK2  PAGES=31.30..............17.16
                END
                 EOS
```

```
          IDENT    FCT
* THIS MODULE GIVES THE DEVICE CORRESPONDING TO A FILE CODE.
* FOR EVERY FILE CODE ,THERE IS AN ADDRESS IN THE DEVICE WORK TABLE
          ENTRY    F CT
*
          EXTRN    D WAS1
          EXTRN    D WAS2
          EXTRN    D WAS3
          EXTRN    D WPTR
          EXTRN    D WPTP
          EXTRN    D WLP
*
F CT      DATA     F CT1-F CT          * NUMBER OF CHARACTERS IN THIS TABLE
*                                      * FCT EXCLUSIVE
          DATA     D WPTR              *01* SOURCE INPUT STANDARD
          DATA     D WLP               *02* LISTING STANDARD
          DATA     D WPTP              *03* PUNCH STANDARD
          DATA     D;WPTR              *04* OBJECT INPUT STANDARD
          DATA     D WAS1              *05* OPERATOR TYPEWRITER
          DATA     D WAS2              *06*  SLOW TAPE READER
          DATA     D WAS3              *07* SLOW TAPE PUNCH
          DATA     D WPTR              *08* PAPER TAPE READER
          DATA     D WPTP              *09* PAPER TAPE PUNCH
          DATA     D WLP               *0A* LINE PRINTER
F CT1     EQU      *-2
          END
```

```
         IDENT    DWT
*****************************************************************************
* THIS MODULE CONTAINS THE WORK TABLE FOR EVERY DEVICE
*****************************************************************************
         ENTRY    D WT
         ENTRY    P DWLG
         ENTRY    D WTEN
         ENTRY    D WAS1
         ENTRY    D WAS2
         ENTRY    D WAS3
         ENTRY    D WPTR
         ENTRY    D WPTP
         ENTRY    D WLP
         ENTRY    C NASR
         ENTRY    C NPTR
         ENTRY    C NPTP
         ENTRY    C NLP
*****************************************************************************
         EXTRN    D RAS1
         EXTRN    D RAS2
         EXTRN    D RAS3
         EXTRN    D RPTR
         EXTRN    D RPTP
         EXTRN    D RLP
*
         EXTRN    I ASR
         EXTRN    I PR
         EXTRN    I PP
         EXTRN    I LP
*****************************************************************************
*
D WT     EQU      *
D WAS1   DATA     'TY'                  *00* TYPEWRITER
         DATA     /0010                 *02* DEVICE ADDRESS
         DATA     80                    *04* REST LENGTH
         DATA     D RAS1                *06* ACTIVATION DRIVER
         DATA     /8000                 *08* SOFTWARE STATUS
         RES      1                     *10* ECB ADDRESS
         RES      1                     *12* CHARACTER ADDRESS
*                                       *12* BUFFER ADDRESS AT BEGINNING
         RES      1                     *14* REQUESTED LENGTH
         RES      1                     *16* EFFECTIVE LENGTH
         RES      1                     *18* ORDER
         RES      1                     *20* RETRY BIT WITH BASIC ORDER
         RES      1                     *22* OUTPUT * WORD TO INPUT
*                                       *22* INPUT * TABULATION TABLE ADDRESS
         RES      1                     *24* CHECKSUM WITH OBJECT ORDER
                                        *24* LINE PRINTER * SAVE LAST CHARACTER O
         RES      1                     *26* OBJECT 4*4* RIGHT OR LEFT
*                                       *26* LINE PRINTER * SAVE CONTROL CODE
         RES      1                     *28*A5
         RES      1                     *30*A6
         DATA     C NASR                *32* CONTROLLER STATUS ADDRESS
         DATA     /8000                 *34* ATTACH
         DATA     I ASR+2               *36* SST SEQUENCE ADDRESS
*****************************************************************************
*
D WAS2   DATA     'TR'                  *00* TAPE READER
         DATA     /0010                 *02*
         DATA     80
         DATA     D RAS2                *06* DRIVER
         DATA     /C000                 *08*
         RES      9
```

```
          RES     2                       *28*30*
          DATA    C NASR                  *32*
          DATA    /8000
          DATA    I ASR+2
*
**********************************************************************
*
D WAS3    DATA    'TP'                    *00* TAPE PUNCH
          DATA    /0010                   *02*
          DATA    80
          DATA    D RAS3                  *06* DRIVER
          DATA    /C000                   *08*
          RES     9
          RES     2                       *28*30*
          DATA    C NASR                  *32*
          DATA    /8000
          DATA    I ASR+2
*
**********************************************************************
*
D WPTR    DATA    'PR'                    *00* H S P R
          DATA    /0020                   *02*
          DATA    80
          DATA    D RPTR
          DATA    /C000
          RES     9
          RES     2                       *28*30*
          DATA    C NPTR
          DATA    /8000
          DATA    I PR+2
*
**********************************************************************
D WPTP    DATA    'PP'                    *00* H S P P
          DATA    /0030
          DATA    80
          DATA    D RPTP
          DATA    /0000
          RES     9
          RES     2                       *28*30*
          DATA    C NPTP
          DATA    /8000                   *30*
          DATA    I PP+2
*
**********************************************************************
*
*
D WLP     DATA    'LP'                    *00* LINE PRINTER
          DATA    /0006
          DATA    136
          DATA    D RLP                   *06* DRIVER
          DATA    /8000
          RES     9
          RES     2
          DATA    C NLP                   *32*
          DATA/8000
          DATA    I LP+2
*
D WT1     EQU     *
P DWLG    EQU     D WT1-D WT
**********************************************************************
*
D WTEN    EQU     *
**********************************************************************
*
C NASR    DATA    /8000
          RES     1
```

```
*
C NPTR   DATA     /8000
         RES      1
*
C NPTP   DATA     /8000
         RES      1
*
C NLP    DATA     /8000
         DATA     -50                    * NUMBER OF LINES IN A PAGE
*****************************************************************************
         END
X ——→     EOS
```

```
          IDENT    INIT
* THIS MODULE IS ENTERED TO LOAD THE MODULES AND INITIALIZE
*                                        THE RUNNING
*                                        LOAD USER (BASE ADDRESS /0800
*                                        SET BUFFER AREA LIMIT
*                                        SET A15
*                                        LOAD USER REGISTERS FROM SAVE AREA
*                                        INITIALIZE USER PCT (LEVEL 62)
*                                        GIVE CONTROL TO USER
          ENTRY    INIT,RINIT
          ENTRY    MAINEX
          ENTRY    RELOAD
*
*
          EXTRN    CVTSBA,CVTSTB,PCT61
          EXTRN    CVTBKA
          EXTRN    CPT
          EXTRN    SOFMA,FILLAB,SCLFG
          EXTRN    INHCP
          EXTRN    C NASR
          EXTRN    C NPTR
          EXTRN    C NPTP
          EXTRN    C NLP
          EXTRN    PFAR
          EXTRN    LDFLAG
          EXTRN    INHST
          EXTRN    MCABFL
*
*
ECBWT     EQU      2
ECBSLC    EQU      4
SAVADR    EQU      -2
STADR     EQU      -4
STATUS    EQU      0
USPSW     EQU      /F800                  LEVEL 62
INIT      CW       A9,CVTBKA
          RF(1)    *+6
RELOAD    LD       A9,CVTBKA SET USER BASE ADDRESS (NORMALLY /0800)
          ST       A9,CVTBKA
          ST       A9,PCT61+SAVADR
RINIT     LDK      A1,0
MAINEX    EQU      RINIT
          WIM      A1
          ST       A1,FILLAB-2
          ST       A1,INHST
          ST       A1,SCLFG
          ST       A1,PCT61+STATUS
          ST       A1,PCT61+ECBWT
          ST       A1,PCT61+ECBSCL
          ST       A1,INHCP
          ST       A1,PFAR
          ST       A1,C NLP+2
          ST       A1,MCABFL
          ST*      A1,CVTSBA              INITIALIZE GET CORE AREA
          LD       A1,CPT+4               SET EX.SYS. MEM. PROTECT MASK
          WMP      A1
          LDK.L    A1,CPT+6
          WM2      A1
          LD       A15,CVTSTB             SET A15 TO STACK BASE
          LDK.L    A1,/8000
          ST       A1,C NASR
          ST       A1,C NPTR
          ST       A1,C NPTP
```

```
                ST          A1,C NLP
                LDK.L       A1,TESTLD
                LDK.L       A2,USPSW
                MSR         2,A15
                RTN         A15
TESTLD          LD          A1,LDFLAG
                RB(0)       TESTLD
                LD          A1,INHST
                RB(0)       TESTLD
                INH
                LD          A1,PCT61+STADR
                LDR         A2,A1
                ANK         A2,1
                ORK.L       A2,USPSW
                MSR         2,A15
                LD          A1,CVTBKA
                LD          A2,CVTSBA
                RTN         A15                 CONTROL TO USER
                END         INIT
                 EOS
```

```
                IDENT    HALTES
                ENTRY    I TC
                ENTRY    I DISK
                ENTRY    I CR
                ENTRY    I CP
                ENTRY    M A00
                ENTRY    I MA00
                ENTRY    I MA01
                ENTRY    I MA02
                ENTRY    I MA03
                ENTRY    I MA04
                ENTRY    I MA05
                ENTRY    I MA06
                ENTRY    I MA07
                ENTRY    I MA08
                ENTRY    I MA09
                ENTRY    I MA10
                ENTRY    I MA11
                ENTRY    I MA12
                ENTRY    I MA13
                ENTRY    I MA14
                ENTRY    I MA15
I MA00          EQU      *
I MA01          EQU      *
I MA02          EQU      *
I MA03          EQU      *
I MA04          EQU      *
I MA05          EQU      *
I MA06          EQU      *
I MA07          EQU      *
I MA08          EQU      *
I MA09          EQU      *
I MA10          EQU      *
I MA11          EQU      *
I MA12          EQU      *
I MA13          EQU      *
I MA14          EQU      *
I MA15          EQU      *
I TC            EQU      *
I DISK          EQU      *
I CR            EQU      *
I CP            EQU      *
M A00           EQU      *
                HLT
                RB(7)    *-2
* STOP COMPUTER IF UNEXPECTED INTERRUPT OR BRANCH
                END
                 EOS
```

```
                IDENT     I_RTN
                ENTRY     I_PFAR
                ENTRY     I_RTC
                ENTRY     I_MEMP
                ENTRY     PFAR
                EXTRN     PCT61
                EXTRN     SYSAB
STATUS          EQU       0
I_PFAR          RIT       /17
                STR       A1,A15
                LDK.L     A1,0
PFAR            EQU       *-2
                RF(1)     AUTRES
                LDR*      A1,A15
                MSR       14,A15
                ST        A15,SAVA15
                IM        PFAR
                HLT                              POWER FAILURE HALT
*
*
AUTRES          LDK       A1,0
                ST        A1,PFAR
                LDK.L     A15,SAVA15
SAVA15          EQU       *-2
                LD        A1,PCT61+STATUS
                ANK       A1,/7F
                RF(4)     ABORT
                MLR       14,A15
                RTN       A15
ABORT           LDK       A2,1
                ADK.L     A15,12
                LD        A3,20,A15
                AB.L      SYSAB
*
*
I_RTC           RIT       /1B
                RTN       A15
I_MEMP          RIT       /1E
                MSR       8,A15
                LDK       A2,3
                LD        A3,20,A15
                AB.L      SYSAB
                END
                 EOS
```

```
          IDENT    EXIT
          ENTRY    EXIT
          EXTRN    DISPAT,SCLFG
          EXTRN    PCT61
          EXTRN    EXSCH
          EXTRN    CVTSBA
* EXIT MODULE FOR GAMMA 4K EXEC.SYST.
STATUS    EQU      0
SAVADR    EQU      -2
STADR     EQU      -4
EXIT      LD       A1,SCLFG
          RF(1)    EXIT1
          ADK      A6,0
          RF(2)    LLEXIT
*                                   EXIT FROM MAIN ,SET EXIT BIT
EXITA     LDK.L    A1,/800
          OR.S     A1,PCT61+STATUS
EXIT2     AB.L     DISPAT
* EXIT FOR SCHEDULE LABEL*************************************************
EXIT1     LDK      A1,0
          ST       A1,SCLFG
          IM       EXSCH             SET EXIT FOR SCHEDULE LABEL FLAG
          RB       EXIT2
************************************************************************
LLEXIT    ST       A9,PCT61+STADR
          SUK      A7,36
          ST       A7,PCT61+SAVADR
          ST       A8,CVTSBA
          LDK.L    A6,-1
          AD.S     A6,PCT61+STATUS
          LDK      A6,0
          RB       EXITA
          END
EOS
```

```
                IDENT   WAIT
* ON ENTRY A8 CONTAINS ECB ADDRESS
                ENTRY   WAIT
                ENTRY   PWAIT
                EXTRN   SCLFG
                EXTRN   DISPAT
                EXTRN   PCT61
ECBWT           EQU     2
ECBSCL          EQU     4
PWAIT           EQU     *
WAIT            LD      A1,SCLFG
                RF(0)   WAIT1
                ST      A8,PCT61+ECBSCL
                RF      RETURN
WAIT1           ST      A8,PCT61+ECBWT
RETURN          AB.L    DISPAT
                END
 EOS
```

```
                IDENT   PCT61
                ENTRY   PCT61
STADR           DATA    0
SAVADR          DATA    0
STATUS          DATA    0
PCT61           EQU     *-2
ECBWT           DATA    0
ECBSCL          DATA    0
                END
 EOS
```

```
          IDENT    GETBUF
*****************************************************************************
* THIS MODULE ALLOCATES DYNAMIC MEMORY BLOCK TO THE USER ON THE
*        SAGITTAIRE GAMMA WITH 4K OF CORE MEMORY
*
*
*                  CALLING SEQUENCE     A7 =BLOCK LENGTH IN CHARACTER
*                                       LKM
*                                       DATA  4
*                  UPON RETURN          OLD A14 IN BLOCK
*                                       NEW A14 GIVES ADDRESS OF THE
*                                       BEGINNING OF THE BLOCK
*
*                           IF NO ROOM   A7=+1
*
*
*
*****************************************************************************
          ENTRY    GETBUF
          EXTRN    CVTMSZ,CVTSBA,CVTBBA
          EXTRN    DISPAT
          EXTRN    CHLEV
          EXTRN    SYSAB
*
*
GETBUF    LDK      A1,48
          ST       A15,SAV15
          CF       A15,CHLEV
          LDR      A8,A6
GET80     ADK      A7,0
          RF(4)    GET1
*
          LD       A7,CVTMSZ          A7=0 GIVE MEMSIZE TO USER (IN AI4)
          RF       RETURN
*
*0  0  0  0  0  0     8     0     0     7    78
GET1      LD       A2,CVTSBA          GET ADDRESS OF 1ST AREA LOCATION
          ADK      A7,5
          ANK.L    A7,/FFFE
          LDK      A6,0
          LDK      A4,1
*
*
GET2      LDR*     A1,A2
          RF(0)    GET5               END OF ALLOCATED AREA,GO AND SEE
*                                     IF THERE IS ENOUGH ROOM LEFT
*
          TM       A1,A4              IS THE BLOCK (ALLOCATED) BUSY
          RF(0)    GET4               NOT BUSY,GO AND SEE IF THE SIZE IS O.K.
*
*
GET3      LDR*     A2,A2              GO TO NEXT BLOCK
          RB(7)    GET2
*
*
GET4      LDR      A5,A1              A5=IND OF BLOCK
          SUR      A5,A2              A5=LENGTH OF BLOCK
*
*
          CWR      A5,A7
          RB(2)    GET3               BLOCK TO SMALL
*
          ADK      A6,0
```

```
                RF(0)   GET4A
                CWR     A6,A5           A6= PREVIOUS AVAILABLE BLOCK LENGTH
                RB(2)   GET3           PREVIOUS IS BETTER
*
GET4A           LDR     A6,A5           BLOCK ALL RIGHT GO TO SEE
                LDR     A3,A2           IF THERE IS ANYTHING BETTER
*
                RB(7)   GET3
*
*
*
*
GET5            ADK     A6,0            HAVE WE FOUND AN ALLOC-NOTBUSY BLOCK
                RF(4)   GET8            YES
                ANK.L   A2,/FFFE
*
                LDR     A1,A2           NO LOOK IF THERE IS STILL ROOM ENOUGH
                ADR     A1,A7
                CW      A1,CVTBBA
                RF(2)   GET6            NOT YET IN OVERFLOW
GET5A           LDK     A7,1            CORE D'FLOW
                RF      RETURN
GET6            CW      A2,CVTSBA
                RF(2)   ERROR           AREA DESTROYED BY USER
                ADK     A1,1            SET BLOCK BUSY
                STR     A1,A2
                LDK     A7,0
                ST*     A7,0,A2         PUT IN NEXT BLOCK LINK
*
GET7            ST      A14,2,A2        PUT OLD AI4 IN BLOCK
                LDK     A7,0
                ADK     A2,6
                LDR     A14,A2          UPDATE NEW AI4
*
*
RETURN          LDR     A6,A8
                LDK.L   A1,SAV15
SAV15           EQU     *-2
                ST      A7,4,A1
                AB.L    DISPAT
*
*
GET8            LDK.L   A5,CVTSBA       CHECK IF ANYTHING
                CWR*    A3,A5           WRONG-
                RF(2)   ERROR
                ADK     A5,2
                CWR*    A3,A5
                RF(6)   ERROR
*
*
                IMR     A3              RESET BUSY-BLOCK FLAG
                LDR     A2,A3
                RB      GET7            GO TO UPDATE A14
*
*
*
ERROR           LDK     A2,4
                LD      A3,SAV15
                LD      A3,20,A3
                AB.L    SYSAB
*
*
                END
 EOS
```

```
          IDENT    FRBUFF
*RELEASING OF A BLOCK IN DYNAMIC BUFFER AREA
*
*                  UPON ENTRY          A14 =USER ENTRY POINT IN THE BLOCK
*                                      A7  =LENGTH OF THE BLOCK
*                  UPON RETURN
*                  UPON RETURN         AM  =0  O.K.
*                                      AM  =1  ERROR IN PARAMETERS
*                                              OR AREA DESTROYED
          ENTRY    FRBUFF
          EXTRN    CVTSBA,CVTBBA
          EXTRN    DISPAT
          EXTRN    CHLEV
*
*
FRBUFF    LDK      A1,48
          LDR      A8,A15
          CF       A15,CHLEV
FREE0     LDR      A2,A14
          SUK      A2,6                A2 =BEGINNING OF THE BLOCK
          CW       A2,CVTSBA
          RF(2)    ERROR               BAD A14 GIVEN
          CW       A2,CVTBBA
          RF(6)    ERROR
*
          ADK      A7,5
          ANK.L    A7,/FFFE
          LDR*     A3,A2
          LDR*     A1,A3
          SUR      A3,A2               A3=LENGTH OF BLOCK
          SUK      A3,1
          CWR      A7,A3
          RF(4)    ERROR
          ADK      A1,0
          RF(0)    FREE1
*
          LDK.L    A4,/FFFE            THE BLOCK IS NOT THE LAST
          ANR.S    A4,A2               OF THE CHAIN , RA2 BUSY BIT
          RF       FREE2
FREE1     STR      A1,A2               END OF THE CHAIN PUT 0
*                                      IN LAST FORWARD LINK
*
*
FREE2     ADK      A2,2
          LDR*     A14,A2              RESER A14 TO USER
*
          LDK      A7,0
RETURN    ST       A7,4,A8
          AB.L     DISPAT
*
ERROR     LDK      A7,1
          RB       RETURN
*
          END
 EOS
```

```
                IDENT    I MHDL
*****************************************************************************
*
*
* INTERRUPT HANDLER ,DO AN INDIRECT BRANCH TO THE INTERRUPT
*         PROCESSOR
*
*****************************************************************************
                ENTRY    INTAB
                ENTRY    I MHDL
*
*
*
                EXTRN    I MA00
                EXTRN    I MA01
                EXTRN    I MA02
                EXTRN    I MA03
                EXTRN    I MA04
                EXTRN    I MA05
                EXTRN    I MA06
                EXTRN    I MA07
                EXTRN    I MA08
                EXTRN    I MA09
                EXTRN    I MA10
                EXTRN    I MA11
                EXTRN    I MA12
                EXTRN    I MA13
                EXTRN    I MA14
                EXTRN    I MA15
INTAB           DATA     I MA00
                DATA     I MA01
                DATA     I MA02
                DATA     I MA03
                DATA     I MA04
                DATA     I MA05
                DATA     I MA06
                DATA     I MA07
                DATA     I MA08
                DATA     I MA09
                DATA     I MA10
                DATA     I MA11
                DATA     I MA12
                DATA     I MA13
                DATA     I MA14
                DATA     I MA15
*
*
I MHDL          MSR      8,A15
                RIL      A1
                LDK      A2,0
                ADK      A1,0
                RF(2)    SWITCH
                CWK      A1,/FF
                RF(1)    SHIFT
                LDK      A2,16
                ECR      A1,A1
                ADK      A1,0
                RF(2)    SWITCH
SHIFT           ADK      A2,2
                SLL1     A1
                RB(6)    SHIFT
SWITCH          ABI      INTAB,A2
*
```

EOS

END

****
****

```
          IDENT    NSCHLB
**********************************************************************
*SCHEDULE LABEL DISPATCHEUR
*          CREATE A NEW ENTRY IN SCHEDULE LABEL DILE IF A6≠0
*         THEN LOOKS FOR LAST LEVEL IN STACK
*                   IF< 48                  RTN AFTER MLR
*                   IF  49<L<61             ERROR HALT
*                   IF  62                  PROCESS LABEL IF ANY
*                   IF  63                  RESET FIRST MAIN REGISTERS, THEN PROCESS
*                                           AS FOR 62.
          ENTRY    DISPAT
          ENTRY FILLAB
          ENTRY    SCLFG
          ENTRY    EXSCH
*
*
          EXTRN    PCT61
          EXTRN    MAINEX
          EXTRN    CHLEV
          EXTRN    SYSAB
*
*
SAVADR    EQU      -2
STATUS    EQU      0
ECBWT     EQU      2
ECBSCL    EQU      4
DISPAT    INH
          ADK      A6,0
          RF(0)    DISP1                  NO LABEL TO PUT IN FILE
          LD       A1,FILLAB-2
          CWK      A1,14
          RF(2)    *+10
          LDK      A2,5
          LDR      A3,A6
          AB.L     SYSAB                  ERROR TO MANY LABEL
          ST       A6,FILLAB,A1           PUT NEW ENTRY IN FILE
*                                         INCREMENT NUMBER OF ENTRY
          ADK      A1,2
          ST       A1,FILLAB-2
          LDK.L    A1,-1
          AD.S     A1,PCT61+STATUS
DISP1     LD       A1,18,A15              TAKE PSW INTERRUPTED
          SRL      A1,10
          SUK      A1,50
          RF(1)    *+6                     49
RETURN    MLR      8,A15                  QUICK BACK TO PROGRAM
          RTN      A15
          SUK      A1,11
          RF(1)    *+4
          HLT                             BIG ERROR 49 LEVEL 62
          SUK      A1,1
          RF(0)    *+6
          ADK.L    A15,20
          LC       A1,PCT61+STATUS
          ANK      A1,/40                 IN PAUSE
          RF(4)    DISP3                  YES
          LD       A2,SCLFG
          RF(0)    DISPAB
          LD       A3,PCT61+ECBSCL        IN WAIT (SCH. LABEL )
          RB(0)    RETURN                 NO
          LDR*     A2,A3                  EVENT OCCURRED
          RF(6)    DISP3                  NO
          ST       A1,PCT61+ECBSCL        YES RAZ EVENT ADDRESS
```

```
          RB        RETURN
DISPAB    LDK.L     A1,0                    EXIT OF SCHEDULED LABEL
EXSCH     EQU       *-2
          RF(5)     DISPX                   NO
          LDK       A1,0                    RESET EXSCH
          ST        A1,EXSCH
          ADK.L     A15,20
          LD        A3,PCT61+SAVADR
          MLR       2,A3
          MSR       2,A15
          ADK       A3,4
          MLR       14,A3
          MSR       8,A15
DISPX     LD        A2,FILLAB-2
          RF(0)     DISP2                   NO ELEMENT IN LABEL FILE
          LD        A1,PCT61+SAVADR
          ST        A14,30,A1
          ADK       A1,4
          LDR       A14,A1
          MLR       8,A15
          MSR       13,A14
          MLR       2,A15
          MS        2,-4,A14
          IM        SCLFG                   SET INTERRUPT SEQUENCE RUNNING FLAG
          LD        A2,FILLAB-2
          LD        A3,FILLAB-2,A2
          STR       A3,A15
          SUK       A2,2
          ST        A2,FILLAB-2
          SUK.L     A15,2
          RTN       A15
DISP2     LD        A3,PCT61+ECBWT          IN WAIT (MAIN)
          RF(0)     DISP2A                  NO
          LDR*      A2,A3                   EV. OCCURRED
          RF(6)     DISP3                   NO
          LDK       A1,0
          ST        A1,PCT61+ECBWT          RAZ EVENT ADDRESS
          RB        RETURN
DISP2A    LD        A1,PCT61+STATUS
          SLC       A1,4                    EXIT BIT ON
          RB(6)     RETURN                  NO,RETURN TO MAIN
*                                           YES IS EVERYTHING QUIET
          ANK.L     A1,/7F0                 /7F0 BECAUSE SCL BEFORE
          RF(4)     DISP3
          LDK       A1,0
          ST        A1,PCT61+STATUS
          LDK       A1,48
          CF        A15,CHLEV
          LDK.L     A8,ECBEX
          LDK       A7,5
          LKM
          DATA      1
          LDR*      A7,A8
          RB(6)     *-2
          AB.L      MAINEX                  GO TO MAINEX
ECBEX     DATA      5
          DATA EXMSG
          DATA      8
          RES       2
EXMSG     DATA      /0D0A
          DATA      'EXIT'
          DATA      /0D0A
*
*
*****************************************************************************
*
```

```
*                                        NOT YET QUIET ,IDLE TASK
DISP3    LDK.L    A2,/FC00
         LDK.L    A1,ADDR
         MSR      2,A15
         RTN      A15
ADDR     RB       ADDR                   IDLE LOOP
         DATA     0
FILLAB   RES      7
SCLFG    DATA     0
         END
 EOS
```

```
            IDENT   M A00
            ENTRY   M A00
            ENTRY   M B00
            EXTRN   MPYMOD
            EXTRN   DIVMOD
            EXTRN   ADDMOD
            EXTRN   DSUMOD
            EXTRN   SYSAB
*                   VALIDITY CHECK OPC
*
* REGISTERS  VALUE
*           A1       USER DATA
*           A2       ADDRESS OF USER DATA
*           A3       USER DATA- 2
*
*
L TOPC      EQU     3                   T OPC LENGTH.-1
L TSVR      EQU     216
*
M A00       LDK     A4,L TOPC           INITIALIZATION OF T OPC POINTER
            LDR     A5,A3               LOAD USER DATA
            SRL     A5,8                OPC RIGHT CHARACTER OF A5
*
M A01       CC      A5,T OPC,A4 TEST OPC USER DATA = OK
            RF(0)   M A02               OK GO TO M A02
            SUK     A4,1
            RB(6)   M A01
            LDR     A3,A2               USER P FOR ABORT ROUTINE
            LDK     A2,2                INVALID OPC ERROR
            AB.L    SYSAB
*
M A02       LDK     A5,L TSVR           INITIALIZATION OF T SVR POINTER
M A03       LD      A8,T SVR,A5         IS THIS T SVR AREA FREE
            RF(0)   M A04               YES GO TO M A04
            SUK     A5,18               NO UPDATE OF T SVR POINTER
            RB(6)   M A03               T SVR OVERFLOW   NO GO TO M A03
            LDK     A2,1                YES SYSTEM ABEND
            AB.L    SYSAB
*
M A04       LDR     A8,A5               LOAD T SVR POINTER
            ADK.L   A8,T SVR+4
            MS      8,Z SVR             SAVE MONITOR REGISTERS
            MLR     8,A15               LOAD USER REGISTERS
            MS*     14,Z SVR+14         SAVE USER REGISTERS IN T SVR
            LDK.L   A8,-4
            AD.S    A8,Z SVR+14
            MLR     2,A15               LOAD USER PSW AND P
            MS*     2,Z SVR+14          USER PSW AND P IN T SVR
            LDK.L   A1,M B00            LOAD NEW P REGISTER
            ANK.L   A2,/FFFE
            MSR     2,A15               NEW PSW IN STACK
            ML      8,Z SVR             LOAD MONITOR REGISTERS
            RTN     A15                 GO TO START M B00 IN USER LEVEL
* THIS MODULE COMPUTE THE SECOND OPERAND OF THE USER DATA,AND
*       CALL A ROUTINE FOR SIMULATE THE USER OPC.
*
M B00       LDR     A6,A3               LOAD USER DATA
            ANK.L   A6,/00DF            IS A CONSTANT INSTRUCTION
            RF(4)   M B01               NO, GO TO M B01
            LD      A6,0,A2
            CWK     A4,2                IS A DAK OPC
            RF(0)   M B00A
            CWK     A4,3                IS A DSK OPC
```

```
          RF(4)     M B00B                NO GO TO M B00B
M B00A    LDK.L     A9,4
          AD.S      A9,T SVR,A5           UPDATE OF USER P
          LD        A7,2,A2
          RF(7)     M B06
M B00B    LDK.L     A9,2
          AD.S      A9,T SVR,A5           UPDATE OF USER P
          RF(7)     M B06
M B01     LDR       A6,A3                 REGISTER NUMBER ANALYSIS (BITS 11 TO 14)
          ANK.L     A6,/0002
          RF(0)     M B01A
          LDR       A6,A3                 BIT 14=1
          SRL       A6,2
          ANK.L     A6,/0007
          ADK       A6,8
          SLL       A6,1                  A6 = 2* REGITER NUMBER
          RF(7)     M B02
M B01A    LDR       A6,A3                 BIT 14=0
          SRL       A6,1
          ANK.L     A6,/000F
*
M B02     LDR       A7,A6                 A7=2 * R2
          ADK       A7,2                  A7=2 * (R2+1)
          LDR       A9,A3                 LOAD USER OPC
          ANK.L     A9,/0040              ADRESSING MODE   2
          RF(4)     M B03                 NO GO TO M B03
*
* REGISTER TO REGISTER INSTRUCTIONS
          ADR       A6,A8                 COMPUTE OPERAND ADDRESS IN T SVR
          LD        A7,4,A6               USER OPERAND BITS 16,31
          LD        A6,2,A6               USER OPERAND (BITS 0 TO 15)
*         USER INDIRECTION TEST.
          LDR       A9,A3                 LOAD USER DATA
          ANK.L     A9,/0020              INDIRECTION
          RF(4)     M B05
          RF(7)     M B06
*
* MEMORY REFERENCE INSTRUCTIONS
*
M B03     LDK.L     A9,2
          AD.S      A9,T SVR,A5 UPDATE OF USER P
*         USER INDEXATION TEST
          LDR       A6,A6
          RF(0)     M B04                 NO GO TO M B04
          ADR       A6,A8
          LD        A6,2,A6               USER INDEXATION
M B04     ADR       A6,A1
*         USER INDIRECTION TEST.
          LDR       A9,A3                 LOAD USER DATA
          ANK.L     A9,/0060
          XRK.L     A9,/0060              INDIRECTION
          RF(4)     M B05
M B04A    LD        A6,0,A6               LOAD USER INDIRECT ADDRESS
M B05     LD        A7,2,A6               LOA USER OPERAND (BITS 16 TO 31)
          LD        A6,0,A6               LOAD USER OPERAND (BITS 0 TO 15)
*
*
M B06     ML        2,T SVR+4,A5          LOAD USER A1,A2
          LDR       A4,A4                 IS A OPC MULTIPLICATION
          RF(4)     M B07                 NO GO TO M B07
          LDR       A7,A2                 USER A2, IN A7
M B07     LDR       A14,A8
          ADK.L     A14,34
          SLL1      A4
          LD        A9,T SRA,A4           NA9)= SIMULATION ROUTINE ADDRESSES
          CFR       A14,A9
```

```
*
                MS      2,T SVR+4,A5        RESULT IN USER A1,A2
                LC      A1,T SVR+32,A5      CR UPDATE
                SC      A1,T SVR+2,A5
                LD      A1,T SVR,A5         LOAD P
                LD      A2,T SVR+2,A5
                ORK     A2,/01              USER INDICATOR =1
                INH                         INHIBIT INTERRUPT.
                MSR     2,A15               USER PSW UPDATED IN STACK
                LDK     A1,0
                ST      A1,T SVR,A5         USER T SVR SAVE AREA FREE
                ML      14,T SVR+4,A5       UPDATE USER REGISTERS
                ENB                         ENABLE INTERRUPT
                RTN     A15                 LINK TO USER PROGRAM
*
*
*
T OPC   DATA    /C0C8
        DATA    /D0D8
*
*
T SRA   DATA    MPYMOD              SIMULATION ROUTINES ADRRESSES
        DATA    DIVMOD
        DATA    ADDMOD
        DATA    DSUMOD
*
*
Z SVR   RES     8                   SAVE AREA
*
*
T SVR   DATA    0                   SAVE AREA USER PSW AND REGISTERS
        RES     17
        DATA    0
        RES     17
        DATA    0
        RES     17
        DATA    0
        RES     17
        DATA    0
        RES     17
        DATA    0
        RES     17
        END
EOS
```

```
          IDENT    MPYMOD
* THIS ROUTINE EXECUTES THE MULTIPLY INSTRUCTION ON THE ALPHA COMPUTER
* CALLING SEQUENCE
*                  LD              A6,ARGI
*                  LDR             A7,A2
*         CF       A14,MPYMOD
* RETURN           RESULT IN A1,A2
*         REGISTERS                A3,A4,A6,A7, ARE CRUSHED
          ENTRY    MPYMOD
MPYMOD    LDK.L    A3,/8000        IF A6=A7/8000 RETURN IN ERROR EXIT
          CWR      A3,A6
          RF(4)    SUITE
          CWR      A3,A7
          RF(0)    END3
          LDR      A6,A7           *A6=X'8000' A7 X'8000'
          LDR      A7,A3           *EXCHANGE A6=A7
SUITE     LDK      A1,0            *INITIALISE
          LDK      A2,0
          LDK      A3,0
          LDK      A4,15
          ADK      A6,0
          RF(0)    END0
          RF(6)    ARG1PS
          ADK      A3,1            *ARG1 NEGATIVE
          C1R      A6,A6
          ADK      A6,1
ARG1PS    ADK      A7,0
          RF(0)    END0
          RF(6)    ARG2PS
          SUK      A3,1            *ARG2 NEGATIVE
          C1R      A7,A7
          ADK      A7,1
          RF(3)    CASSPE
ARG2PS    SRC      A7,1
          RF(6)    NOVER1
          ADR      A1,A6
NOVER1    SRA      A2,1
          SRC      A1,1
          RF(6)    NOVER2
          ORK.L    A2,/4000
          ANK.L    A1,/7FFF
NOVER2    SUK      A4,1
          RB(4)    ARG2PS
NOVER3    ADK      A3,0                 *END* RESTORE SIN
          RF(0)    END1
          C1R      A1,A1
          C1R      A2,A2
          ADK      A2,1
* CAUTION THE NEXT INSTRUCTION IS ABSOLUTELY NECESSARY TO SET CR
* WHEN THE ADK A2,1 SET AN OVERFLOW CR EX (RESULT=8000)
          ADK      A2,0
          RF(2)    NOVER4
          ADK      A1,1
NOVER4    ANK.L    A2,/7FFF
          RF(7)    END2
END3      ADK      A3,/81
END2      ADK      A3,/81
END1      ADK      A3,/81
END0      SLL      A3,1
          ANK.L    A3,/0300
          LDK.L    A4,/FCFF
          AN.S     A4,2,A14
          OR.S     A3,2,A14             UPDATE CR IN PSW CALLING PROGRAM
```

```
                RTN     A14
CASSPE          LDR     A1,A6           *A7=X'8000' THEN A4=1
*                                       *AND  A1 = A6
                RB(7)   NOVER3
                END
 EOS
```

```
          IDENT    DIVMOD
* THIS ROUTINE EXECUTES THE DIVIDE INSTRUCTION ON THE ALPHA COMPUTER
* CALLING SEQUENCE
*                  A1-A2= DIVIDEND
*                  A6   = DIVISOR
*         CF       A14,DIVMOD
*RETURN*           RESUT IN A1-A2   A1= REMAINDER A2= QUOTIENT
* CAUTION   REGISTERS A3,A4,A6,A7,A8 ARE CRUSHED.
          ENTRY    DIVMOD
DIVMOD    LDR      A8,A1                  SAVE DIVIDEND SIGN
          LDK      A3,0
          ADK      A6,0
          RF(0)    END3                   * IF DIVISOR IS NULL,OVERFLOW
          ADK      A1,0
          RF(4)    SIGNE
          ANK.L    A2,/7FFF
          RF(0)    END0                   * IF DIVIDEND IS NULL, CR=0
* THIS SEQUENCE GIVES THE OPERAND A POSITIVE VALUE
SIGNE     LDK      A7,0
          LDK      A4,16
          ADK      A1,0                   * SIGN OF DIVIDEND
          RF(6)    DIVENP
          ADK      A3,1                   * DIVIDEND IS NEGATIVE
          C1R      A1,A1
          C1R      A2,A2
          ANK.L    A2,X'7FFF'
          ADK      A2,1
          RF(1)    DIVENP
          ADK      A1,1
          RF(5)    END3
DIVENP    ADK      A6,0                   * SIGN OT DIVISOR
          RF(6)    DIVORP
          SUK      A3,1                   * NEGATIVE DIVISOR
          C1R      A6,A6
          ADK      A6,1
          ADK      A6,0                   RESET IF OVERFLOW
          RF(5)    CASSPE
DIVORP    CWR      A1,A6
          RF(1)    END3
          RF(0)    CASSP1
SHIFT     SUK      A4,1                   * A1 LESS THAN A6
          RF(0)    END
          SLL      A1,1
          SLL      A7,1
          SLL      A2,1
          RF(6)    *+4
          ADK      A1,1
          ADK      A1,0
          RF(2)    CASSP1
          CWR      A1,A6
          RB(2)    SHIFT
CASSP1    SUR      A1,A6
          ADK      A7,1
          RB(7)    SHIFT
* THIS SEQUENCE RESTORES THE SIGN OF THE RESULTS
END       LDR      A2,A7
          ADK      A3,0
          RF(0)    NOSIGN
          RF(2)    AA                     DIVIDEND+. DIVISOR+
          C1R      A1,A1                  REMAINDER
          ADK      A1,1
AA        C1R      A2,A2                  * QUOTIENT
          ADK      A2,1
```

```
* CAUTION THE NEXT INSTRUCTION IS ABSOLUTELY TO SET CR
* WHEN THE ADK A2,1 SET AN OVERFLOW CR EX NQUOTIENT=8000)
          ADK       A2,0
          RF(2)     END2
END3      ADK       A3,/81
END2      ADK       A3,/81
END1      ADK       A3,/81
END0      SLL1      A3
          ANK.L     A3,/0300
          LDK.L     A4,/FCFF
          AN.S      A4,2,A14
          OR.S      A3,2,A14            UPDATE CR IN PSW CALLIG PROGRAM
          RTN       A14
NOSIGN    ADK.L     A8,0                TEST OF DIVIDEND SIGN
          RF(6)     BB
          C1R       A1,A1               DIVIDEND-. DIVISOR-
          ADK       A1,1                REMAINDER-
BB        ADK       A2,0
          RB(0)     END0
          RB(2)     END2
          RB(7)     END1
CASSPE    LDR       A7,A1
          LDR       A1,A2
          CWK       A1,/8000
          RB(4)     END
          LDK       A1,0
          RB(7)     END
          END
 EOS
```

```
          IDENT    ADDMOD
* THIS ROUTINE EXECUTES THE DOUBLE ADDITION INSTRUCTION ON THE ALPHA
* COMPUTER
* CALLING SEQUENCE
*                  A1-A2 =ARG1
*                  AG-A7 = ARG2
*         CF       A14,ADDMOD
*RETURN            RESULT IN A1-A2
* CAUTION REGISTERS A3,A6,A7 ARE CRUSHED.
          ENTRY    ADDMOD
ADDMOD    LDK      A3,0
          ANK.L    A7,/7FFF
          ANK.L    A2/7FFF
          ADR      A2,A7
          RF(3)    OVER1
NOVER     ADR      A1,A6                  * NO OVERFLOW
          RF(1)    END1
          RF(2)    END2
          RF(3)    END3
NOVER1    ADK      A2,0
          RF(0)    END0
          RF(7)    END1
OVER1     ANK.L    A2,/7FFF
          ADK      A1,1
          RF(3)    OVER3
          RB(7)    NOVER
OVER3     ADR      A1,A6
          RF(2)    END3
          ADK      A1,0
          RF(1)    END1
          RB(7)    NOVER1
END3      ADK      A3,/80
END2      ADK      A3,/80
END1      ADK      A3,/80
END0      SLL1     A3
          ANK.L    A3,/0300
          LDK.L    A4,/FCFF
          AN.S     A4,2,A14
          OR.S     A3,2,A14          UPDATE CR IN PSW CALLING PROGRAM
          RTN      A14
          END
 EOS
```

```
          IDENT    DSUMOD
* THIS ROUTINE EXECUTES THE DOUBLE SUBSTRACTON INSTRUCTION ON THE ALPHA COMPUTER
* CALLING SEQUENCE
*                  A1-A2  =ARG1
*                  A6-A7  ARG2
*         CF       A14,DSUMOD
* RETURN           RESULT IN A1-A2
* CAUTION REGISTERS A3,A6,A7,  ARE CRUSHED.
          ENTRY    DSUMOD
DSUMOD    LDK      A3,0
          CR1R     A6,A6
          CR1R     A7,A7
          ANK.L    A7,X'7FFF'
          ANK.L    A2,/7FFF
          ADK      A7,1
          RF(1)    NOVER1
          ADK      A6,1
          RF(3)    OVER4
          RF(7)    NOVER
NOVER1    ADR      A2,A7
          RF(3)    OVER1
NOVER     ADR      A1,A6
NOVER2    RF(1)    END1
          RF(2)    END2
          RF(3)    END3
          ADK      A2,0
          RF(0)    END0
          RF(7)    END1
OVER1     ANK.L    A2,X'7FFF'
          ADK      A1,1
          RF(3)    OVER4
          RB(7)    NOVER
OVER4     ADR      A1,A6
          RF(2)    END3
          ADK      A1,0
          RB(7)    NOVER2
END3      ADK      A3,/80
END2      ADK      A3,/80
END1      ADK      A3,/80
END0      SLL1     A3
          ANK.L    A3,/0300
          LDK.L    A4,/FCFF
          AN.S     A4,2,A14
          OR.S     A3,2,A14          UPDATE CR IN PSW CALLING PROGRAM
          RTN      A14
          END
 EOS
```

```
          IDENT    IORM
*
*************************************************************************
*         ENTRY CONDITIONS
*         A6 = SCHEDULE LABEL ADDRESS
*         A7 = ORDER
*         A8 = ECB ADDRESS
*         ENTRY M IORM
*************************************************************************
          ENTRY    M IORM
          ENTRY    E FECB
          ENTRY    E SECB
*
          EXTRN    E S000
          EXTRN    E S015
          EXTRN    E S012
          EXTRN    E S011
          EXTRN    F CT
          EXTRN    L VCH
          EXTRN    PCT61
          EXTRN    DISPAT
          EXTRN    PWAIT
* THIS SEQUENCE PERFORMS THE GET ASSIGN
ORD30     LDR*     A1,A8                *A1=ECB0
          ANK      A1,/FF               *A1=FILE CODE
          RF(0)    ORD301               *FILE CODE=0
          ADR      A1,A1
          CW       A1,F CT
          RF(1)    ORD301               *FILE CODE NOT IN TABLE
          LD       A4,F CT,A1
          RF(0)    ORD301
          LDR*     A1,A4
          LD       A2,4,A4              *LENGTH
          LD       A3,2,A4              *ADDRESS
ORD302    MS       3,2,A8
          LDK      A1,/80
          SCR      A1,A8
          LDK      A1,0
          ST       A1,8,A8
          AB.L(7)  DISPAT
*
ORD301    LDK      A1,0                 *FILE CODE UNKNOWN
          LDK      A2,0
          LDK      A3,0
          RB(7)    ORD302
* BUFFERS TO OUTPUT EOS OR EOF
          DATA     /0A0A
E SECB    DATA     ' EOS'
          DATA     /0D0A
E FECB    DATA     ' EOF'
          DATA     /0D0A
* OUTPUT SEQUENCE
ERS01     AB.L(7)  E S011               * FUNCTION UNKNOWN
NOTAS1    AB.L(7)  E S015
NOTAS2    AB.L(7)  E S000
* THIS SEQUENCE SEARCHES THE DWT CORRESPONDING TO FILE CODE
M IORM    IM       PCT61                * INCREMENT THE EVENT COUNT
          LDK.L    A1,*+8
          AB.L(7)  L VCH                * AT RETURN LEVEL = 48
          ANK.L    A8,/FFFE
          LDR      A1,A7                * IS IT ORDER 30
          ANK      A1,/3F
          CWK      A1,/30
```

```
          RB(0)    ORD30
          LDR*     A1,A8              * A1=ECB0
          ANK      A1,/FF             * A1=FILE CODE
          RB(0)    NOTAS2             * IF FILE CODE =0, GO TO NOTAS
          ADR      A1,A1
          CW       A1,F CT
          RB(1)    NOTAS1             * THE FILE CODE IS NOT IN THE TABLE
          LD       A3,F CT,A1         * A3 = DWT ADDRESS
          RB(0)    NOTAS2             * THE FILE CODE IS NOT ASSIGNED
* THIS SEQUENCE CHECKS THAT THE CONTROLLER IS FREE
          LDR      A4,A8              * SAVE A8
          LD       A8,32,A3           * A8=CONTROLLER STATUS ADDRESS
IORM21    LDR*     A2,A8
          RF(2)    IORM20             * CONTROLLER FREE
          LD       A2,18,A15          * CHECK IF LEVEL   62
          SRL      A2,10
          SUK      A2,62
          RB(2)    IORM21
          LD       A2,20,A15          * RESTORE LKM INSTRUCTION
          LD       A1,PCT61
          SUK      A2,4
          SUK      A1,1
          ADK      A6,0
          RF(0)    IORM22
          SUK      A2,2               * SCHEDULE LABEL
          SUK      A1,1
IORM22    ST       A2,20,A15
          ST       A1,PCT61
          AB.L(7)  PWAIT
IORM20    LDR      A8,A4
* THIS SEQUENCE PUTS THE ECB PARAMETERS IN THE PARAMETERS TABLE
IORM2     LDK      A2,0               * SET BUSY
          ST       A6,30,A3
          LDR      A6,A3
          SCR      A2,A8              * ECB
          SC*      A2,32,A6           * CONTROLLER
          ST       A2,26,A6           * CHECKSUM
          ST       A2,24,A6           * CHARACTER FLAG
          ST       A2,22,A6           * TABULATIOM
          ST       A2,18,A6           * ORDER
          ST       A2,8,A8            * USER STATUS
          LD       A2,4,A8
          AB.L(0)  E SO12             * LENGTH = 0
* THIS SEQUENCE ANALYSES THE ORDER
IORM3     ST       A8,10,A6
          LD       A1,2,A8
          AB.L(0)  E SO12             * NO BUFFER ADDRESS
          LDR      A4,A7
          ANK      A4,/3F
          RB(0)    ERSO1              * ORDER = 0
          CWK      A4,2
          RF(0)    ITAB               * TABULATION
          RF(2)    SWITCH
          CWK      A4,5
          RB(2)    ERSO1              * ORDER = 3OR4
          CWK      A4,6
          RF(0)    REMOVE
          RF(2)    SWITCH
          CWK      A4,9
          RF(2)    SWITC1
          CWK      A4,/14
          RF(0)    IEOS
          CWK      A4,/16
          RF(0)    IEOF
          CWK      A4,/26
          RF(0)    OEOS
```

```
            CWK     A4,/22
            RF(0)   OEOF
            CWK     A4,/31
            RB(2)   ERSO1
            CWK     A4,/38
            RB(1)   ERSO1
            RF      SWITCH
OEOF        LDK.L   A1,E FECB          * OUTPUT EOF
            RF(7)   OEOS+4
*
OEOS        LDK.L   A1,E SECB          * OUTPUT EOS
            LDK     A2,4
            LDK     A4,6
            RF(7)   SWITC1
*
IEOS        LDK.L   A4,/8302           * SKIP TO EOS
            RF(7)   SWITC1
*
IEOF        LDK.L   A4,/8102           * SKIP TO EOF
            RF(7)   SWITC1
ITAB        LD      A5,10,A6
            LD      A5,10,A5
            ST      A5,22,A6
            ANK     A4,/F
            RF(7)   SWITC1
*
* THIS SEQUENCE REMOVES THE TRAILING BLANKS
REMOVE      ADR     A1,A2
            SUK     A1,1
            LCR     A3,A1
            ADK     A1,1
            ANK     A3,/FF
            SUK     A3,/20
            RF(4)   REMOV1
REMOV2      SUK     A1,2
            LDR*    A3,A1
            SUK.L   A3,/2020
            RF(4)   REMOV1
            SUK     A2,2
            RB(4)   REMOV2
            ADK     A2,2
REMOV1      LD      A1,2,A8
            RF(7)   SWITC1
* SWITCH TO SPECIFIC MODULE
SWITCH      LDR     A5,A7
            ANK     A5,/40             * KEEP STATUS BIT
SWITC2      LDK     A3,0
            CWK     A4,7
            RF(2)   SWIBIS
            LD*     A2,2,A8
            ANK     A2,/FF
            ADR     A2,A2
            ADK     A2,2
SWIBIS      MS      5,12,A6            * INITIALISE THE PARAMETERS TABLE
            ANK     A4,/FF
            ABI(7)  6,A6               * SWITCH
SWITC1      LDK     A5,0
            RB(7)   SWITC2
            END
 EOS
```

```
                IDENT   DRIYO1
                ENTRY   D RAS1
                ENTRY   D RAS2
                ENTRY   D RAS3
                ENTRY   I ASR
                EXTRN   C INPT
                EXTRN   C NASR
X ⟶ EXTRN       C WAIT
                EXTRN   D WAS1
                EXTRN   D WAS2
                EXTRN   D WAS3
                EXTRN   E SECB
                EXTRN   F FECB
                EXTRN   E SO11
                EXTRN   I NPUT
                EXTRN   L VCH
                EXTRN   O TPUT
                EXTRN   R TUR1
                EXTRN   R TUR2
                EXTRN   R TUR4
                EXTRN   R TURN
*
S               EQU     1
H               EQU     0
TY01            EQU     /0010
*
* THIS SEQUENCE ACTIVATES THE ASR KEYBOARD IN INPUT OR OUTPUT
D RAS1          LDK     A5,/81
                SC      A5,C NASR+1
                ST      A6,C NASR+2 * DWT ADDRESS
                SUK     A4,5
                RF(2)   DRASIN
                RF(0)   DRASO3
                LD      A4,12,A6
                CWK     A4,E SECB       * IS IT EOS
                RF(0)   **+8
                CWK     A4,E FECB       *IS IT EOF
                RF(4)   DRASO2
                LDK     A4,6
                ST      A4,14,A6
                LDK     A4,5
                ST      A4,18,A6
                RF(7)   DRASO3
DRASO2          EQU     *
                LDK.L   A2,/FFFE        MAKE EVEN THE BUFFER ADDRESS
                AN.S    A2,12,A6
                LD*     A2,12,A6        CONVERT THE FORMAT CONTROL CODE
                LDK     A2,2
                AD.S    A2,12,A6
X ⟶ AD.S        A2,16,A6
                ANK     A4,/FE
                CWK     A4,/30
                RF(4)   DRASO3
                LDK     A1,/0A
                RF(7)   DRASO1
DRASO3          LDK     A1,0
                RF(7)   DRASO1
DRASOT          INH
                CF      A15,O TPUT
                LD      A1,22,A6
DRASO1          LDK     A2,0
                INH
                CIO     A2,S,TY01
```

```
          OTR       A1,0,TY01
          CF        A15,0 TPUT
WAITTS    AB.L(7)   C WAIT
*************************************************************************
DRASIN    LDK       A2,1
          CIO       A2,S,TY01
          RB(7)     WAITTS
* THIS SEQUENCE ACTIVATES THE ASR READER
D RAS2    LDK       A5,/82
          SC        A5,C NASR+1
          ST        A6,C NASR+2 *DWT ADDRESS
          SUK       A4,4
          AB.L(1)   E SO11
          LDK       A1,/11              * X-ON
          LDK       A2,0
          INH
          CIO       A2,S,TY01
          OTR       A1,0,TY01
          CIO       A2,H,TY01
          ENB
          RB(7)     WAITTS
* THIS SEQUONCE ACTIVATES THE ASR PUNCH
D RAS3    LDK       A5,/81
          SC        A5,C NASR+1
          ST        A6,C NASR+2         * DWT ADDRESS
          SUK       A4,5
          AB.L(2)   E SO11
          RB(0)     DRASOT
          SUK       A4,3
          RF(2)     *+8
          LDK       A4,7
          SC        A4,19,A6
          LDK       A1,/12              * TAPE ON
          RB(7)     DRASO1
* THIS SEQUENCE PERFORMS THE ASR INTERRUPT
I ASR     MSR       8,A15
          LD        A1,C NASR
          ANK       A1,/FF
          LD        A6,C NASR+2         * DWT ADDRSSS
          CWK       A1,/82
          RF(4)     I ASR1
          SST       A2,TY01
          LDK       A2,/81
          SC        A2,C NASR+1
          LDK       A2,1
          CIO       A2,S,TY01
          MLR       8,A15
          RTN       A15
I ASR1    SST       A2,TY01
          AB.L(4)   C INPT              * SST REFUSED
          LDK.L     A1,*+8
          AB.L(7)   L VCH
          LD        A1,20,A6            * TEST STATUS BIT
          AB.L(0)   R TUR2
          AB.L(7)   R TUR4
          END
 EOS
```

```
            IDENT    DRP018
*
* THIS MODULE CONTAINS SPECIFIC OPERATIONS FOR PAPER PUNCH
            ENTRY    D RPTP
            ENTRY    I PP
*
            EXTRN    D WPTP
            EXTRN    C OUT
            EXTRN    CWAIT  C:WAIT
            EXTRN    E SO11
            EXTRN    L VCH
            EXTRN    M RETR
            EXTRN    O TPUT
            EXTRN    R TUR4
*
S           EQU      1
H           EQU      0                    PP18  EQU 130
*
* THIS SEQUENCE ACTIVATES THE HIGH SPEED PAPER TAPE PUNCH
* ENTRY CONDITIONS
*           A6 =DWT ADDRESS
*           A4 =ORDER
*
D RPTP      SUK      A4,5                 * CHECK ORDER
            AB.L(2)  E SO11
            INH
            CF       A15,O TPUT
            CIO      A2,S,PP18            * ACTIVATE THE HIGH SPEED PAPER TAPE PUNC
            AB.L(7)  C WAIT
* THIS SEQUENCE PERFORMS THE PTP INTERRUPT
I PP        MSR      8,A15
            LDK.L    A6,D WPTP
            SST      A2,PP18
            AB.L(4)  C OUT
            LDK.L    A1,*+8
            AB.L(7)  L VCH
            ANK      A2,/FF
            RF(0)    I PP1
            LD       A3,20,A6             * TEST STATUS BIT
            RF(1)    I PP2
            LDK      A1,0                 * GO TO M RETR
            LDR      A4,A2
            ANK      A4,/1
            RF(1)    I PP3
            LDK      A1,1
I PP3       AB.L(7)  M RETR
I PP2       ORK.L    A2,/8000             * HARDWARE STATUS
I PP1       AB.L(7)  R TUR4               * END OF IO
            END
 EOS
```

```
          IDENT    DRPR38
*
* THIS MODULE CONTAINS SPECIFIC OPERATIONS FOR PAPER READER
          ENTRY    D RPTR
          ENTRY    I PR
*
          EXTRN    D WPTR
          EXTRN    C INPT
          EXTRN    C WAIT
          EXTRN    E SO11
          EXTRN    L VCH
          EXTRN    M RETR
          EXTRN    R TUR2
          EXTRN    R TUR4
*
S         EQU      1
H         EQU      0
PR38      EQU      /0020
*
* THIS SEQUENCE ACTIVATES THE HIGH SPEED PAPER TAPE READER
* ENTRY CONDITIONS
*         A6 = DWT ADDRESS
*         A4 = ORDER
*
D RPTR    SUK      A4,4                 * CHECK ORDER
          AB.L(1)  E SO11
          CIO      A2,S,PR38            * ACTIVATE  HIGH SPEED PAPER READER
          AB.L(7)  C WAIT
* THIS SEQUENCE PERFORMS THE PTR INTERRUPT
I PR      MSR      8,A15
          LDK.L    A6,D WPTR
          SST      A2,PR38
          AB.L(4)  C INPT
          LDK.L    A1,*+8
          AB.L(7)  L VCH
          ANK      A2,/FF
          AB.L(0)  R TUR2               * HARDWARE STATUS = 0
          LD       A3,20,A6             * TEST STATUS BIT
          RF(1)    I PR1
          LDK      A1,0
          AB.L(7)  M RETR
I PR1     ORK.L    A2,/8000
          AB.L(7)  R TUR4               * HARDWARE STATUS   0
          END
 EOS
```

```
          IDENT    DRLP
*
          ENTRY    D RLP
          ENTRY    I LP
*
          EXTRN    C NLP
          EXTRN    C WAIT
          EXTRN    D WLP
          EXTRN    E SECB
          EXTRN    E FECB
          EXTRN    E SO11
          EXTRN    L VCH
          EXTRN    M RETR
          EXTRN    R TUR1
          EXTRN    R TUR4
*
*THIS SEQUENCE PREPARES THE MULTIPLEX AND ACTIVATES IT
*
S         EQU      1
LP        EQU      /0006
MULTI     EQU      /0098
*
*         A4 = ORDER
*         A6 V DWT ADDRESS
MULTIC    RES      2
*
D RLP     SUK      A4,5              *CHECK ORDER
          AB.L(2)  E SO11
          SUK      A4,1
          RF(0)    DRLP02
          AB.L(1)  E SO11
DRLP1B    LD       A2,12,A6
          LD       A1,14,A6
DRLP01    ANK.L    A1,/FFF
          ADR      A2,A1
          SUK      A2,1
          C1R      A1,A1
          ADK      A1,1
          ANK.L    A1,/FFF
          ORK.L    A1,/8000          *FUNCTION CHARACTER OUTPUT
          MS       2,MULTI
          MS       2,MULTIC
          CIO      A2,S,LP           * ACTIVATE
          AB.L(7)  C WAIT            * EN DOF ACTIVATION
*
DRLP02    LD       A4,12,A6
          CWK      A4,E SECB         * IS IT EOS
          RF(0)    *+8
          CWK      A4,E FECB         *IS IT EOF
          RF(4)    DRLP2B
          SUK      A4,1
          ST       A4,12,A6
          LDR*     A1,A4             * SAVE CONTROL CODE
          ST       A1,26,A6
          LDK      A4,6
          ST       A4,14,A6
          IM       C NLP+2           * NUMBER OD LINES +1
          LDK      A4,5
          ST       A4,18,A6
          RB(7)    DRLP1B
DRLP2B    LD       A5,12,A6          * PUT A CR-LF EACH N CHARACTERS
          LD       A4,4,A6
          LD       A3,14,A6
```

```
                LDK.L   A2,/0A0D
                ANK.L   A3,/FFF
DRLP04          CWR     A3,A4
                RF(5)   DRLP03
                ADR     A5,A4
                SUR     A3,A4
                STR     A2,A5
                IM      C NLP+2         *NUMBER OD LINES +1
                RB(7)   DRLP04
*
DRLP03          ADR     A5,A3
                LCR     A1,A5
                SC      A1,24,A6        SAVE LAST CHARACTER OF BUFFER
                SCR     A2,A5
                LD      A5,12,A6
                LDR*    A1,A5           *SAVE THE CONTRL CODE
                ST      A1,26,A6
                CWK     A1,/31
                RF(0)   PAGE            *CONVERT THE CONTROL CODE
                CWK     A1,/30
                RF(0)   TWOLIN
                CWK     A1,/2B
                RF(0)   SUPERP
                LDK.L   A2,/0D0A        *ONE LINE
                IM      C NLP+2         *NUMBER OF LINES +1
                RF(6)   PAGE
                RF(7)   DRLP05
*
SUPERP          LDK.L   A2,/0D0D        *SUPERPOSITION
                RF(7)   DRLP05
TWOLIN          LDK.L   A2,/0A0A        *SKIP TWO LINES
*
                IM      C NLP+2         *NUMBER OD LINES +1
                IM      C NLP+2         *NUMBER OD LINES +1
                RF(2)   DRLP05
*
PAGE            LDK.L   A2,/0D0C        *SKIP TO TOP OF PAGE
                LDK.L   A1,-50
                ST      A1,C NLP+2
DRLP05          STR     A2,A5
                IM      14,A6
                RB(7)   DRLP18
*
*****************************************************************************
*
I LP            MSR     8,A15
                SST     A2,LP
                LDK.L   A6,D WLP
                LDK.L   A1,*+8
                AB.L(7) L VCH
                ANK     A2,/FF          *TEST STATUS
                RF(0)   ENDLP2
                LD      A1,20,A6        *TEST RETRY BIT
                RF(1)   ENDLP1
                LDR     A3,A2
                ML      2,MULTIC        *RESTORE MULTIPLEX
                MS      2,MULTI
                LDR     A2,A3
                LDK     A1,0
                LDK     A3,0            *CO TO RETRY
                AB.L(7) M RETR
*END OF IO
ENDLP1          ORK.L   A2,/8000
ENDLP2          LD      A4,26,A6        *RESTORE CONTROL CODE
                LD      A5,12,A6        BUFFER ADDRESS
                STR     A4,A5
```

```
                LD      A1,18,A6        ORDER
                SUK     A1,6
                RF(2)   ENDLP3
                LC      A1,24,A6
                LD      A3,14,A6
                SUK     A3,1
                ADR     A3,A5
                SCR     A1,A3
ENDLP3          AB.L(7) R TUR4
                AB.L(7) R TUR4
                END
 EOS
```

```
          IDENT    DRCR
*
          ENTRY    D RCR
          ENTRY    I CR
*
          EXTRN    C WAIT
          EXTRN    D WCR
          EXTRN    E SO11
          EXTRN    L VCH
          EXTRN    M RETR
          EXTRN    R TUR1
          EXTRN    R TUR3
*
CR1       EQU      /0005
CRMULT    EQU      /0094
H         EQU      0
S         EQU      1
*
*****************************************************************************
*
* THIS SEQUENCE CHECKS THE PARAMETERS,PREPARES AND ACTIVATES THE MULTIPLEX
*
D RCR     SUK      A4,2                  * CHECK ORDER
          AB.L(1)  E SO11
          LDK.L    A2,CRBUFF+158
          LDK.L    A1,/4F60
          MS       2,CRMULT
          CIO      A2,S,CR1
          AB.L(7)  C WAIT
*
* SYSTEM BUFFER
CRBUFF    RES      80
*
STATUS    RES      1
*
* THIS SEQUENCE IS ENTERED BY AN INTERRUPT
*
* CHECK IF STATUS IS NULL
I CR      MSR      8,A15                 * SAVE REGISTERS
          LDK.L    A6,D WCR
          SST      A2,CR1                * SEND STATUS
          LDK.L    A1,*+8
          AB.L(7)  L VCH
          LDK      A1,0
          ST       A1,STATUS
* THIS SEQUENCE SWITCHES BY ORDER
          LD       A1,18,A6
          ANK      A1,/FD                * ORDER 2
          AB.L(4)  ITCR5                 * NO
          ANK      A2,/F                 * KEEP USEFUL BITS
          RF(0)    ITCR11
ITCR2     LDK.L    A1,/4F60
          LDR      A3,A2
          LDK.L    A2,CRBUFF+158
          MS       2,CRMULT              * MULTIPLEX
          LDR      A2,A3
* GO TO RETRY MODULE
RETRY     LDK      A1,0
          LDK      A3,0
          AB.L(7)  M RETR
ITCR11    LDK.L    A5,CRBUFF
          LDK      A7,0
          LD       A8,14,A6
```

```
                LD      A4,12,A6
HOLCR8          LDR*    A1,A5           * TRANSLATION
                RF(1)   HOLCR1
                LDK     A3,/20          * H-CODE=0 .. A-CODE = 20
                RF(7)   HOLCR9
*
HOLCR1          LDK     A3,0            * SCAN FOR THE FIRST HOLE
HOLCR2          ADK     A3,1
                SLL     A1,1
                RB(1)   HOLCR2
                SUK     A3,4
                ANK.L   A1,/7FFF        * FIRST HOLE
                RF(4)   HOLLET
                SUK     A3,1            * SWITCH
                RF(1)   *+12
                RF(0)   *+6
                LDK     A3,/26          * $
                RF(7)   HOLCR9          * STORE
                LDK     A3,/2D          * -
                RF(7)   HOLCR9          * STORE
                ADK     A3,/2F          * DIGIT
                RF(7)   HOLCR9          * STORE
*
HOLFST          RES     1
HOLSND          RES     1
*
HOLLET          ST      A3,HOLFST       * SAVE FIRST HOLE POSITION
                ADK     A3,1
                SLL     A1,1
*
                RB(1)   *-4
                ST      A3,HOLSND       * SAVE SECOND HOLE POSITION
                ANK.L   A1,/7FFF
                RF(4)   HOLBIZ
                LD      A1,HOLFST       * SWITCH
                CWK     A1,3
                RF(6)   HOLLE1
                ADR     A1,A1           * FIRST HOLE = 12-11=0
                LD      A1,TABLE1,A1    * CHOOSE THE TABLE
                LD      A2,HOLSND
*
                SUK     A2,3
                RF(2)   HOLCR3
                ADR     A1,A2
                LCR     A3,A1           *CHOOSE THE LETTER IN THE TABLE
                RF(7)   HOLCR9
HOLLE1          RF(0)   HOLCR3
                SUK     A1,4
                CWK     A1,5
                RF(1)   HOLCR3
                LD      A2,HOLSND
                SUK     A2,10
                RF(4)   HOLCR3
                LC      A3,TABL1,A1     *CHOOSE CHARACTER IN THE TABLE
                RF(7)   HOLCR9          *STORF
*
TABL1           DATA    /3A23
                DATA    /4027
                DATA    /3D22
*
HOLBIZ          LD      A3,HOLFST       *THREE HOLES

                SUK     A3,2
                RF(1)   HOLCR3
                LD      A3,HOLSND
                SUK     A3,4
```

```
                RF(2)     HOLCR3
                SUK       A3,5
                RF(1)     HOLCR3
                ADK       A3,9                *THIRD HOLE = HOLE EIGHT
                ADK       A3,1
                SLL       A1,1
                RB(1)     *-4
                SUK       A3,10
                RF(4)     HOLCR3
                SLL       A1,1
                RF(4)     HOLCR3
                LD        A1,HOLFST
                ADR       A1,A1
                LD        A1,TABLE2,A1
                AD        A1,HOLSND
                SUK       A1,4
                LCR       A3,A1
                RF(7)     HOLCR9
*STOR           E ASCII   CODE
HOLCR9          SCR       A3,A4
                ADK       A4,1
                ADK       A5,2
                ADK       A7,1
                CWR       A7,A8
                RB(4)     HOLCR8
                ST        A7,16,A6
*
*THIS SEQUENCE PERFORMS THE END OF TRANSFERT
HOLEND          LD        A1,STATUS
                ANK       A1,/1
                RF(0)     HOLEN2
                LDK       A2,/4               *GO TO RETRY WITH DATA FAULT
                AB.L(7)   ITCR2
HOLEN2          ST        A4,12,A6
                SUR       A4,A7               BUFFER ADDRESS
                MLR       2,A4
                CWK       A1,/3A45            * E
                RF(4)     HOLEN1
                CWK       A2,/4F46            *OF
                RF(0)     HOLEN0
                CWK       A2,/4F53            *OS
                RF(4)     HOLEN1
HOLEN0          ADK       A4,4
                ST        A4,12,A6
                LDK       A4,4
                ST        A4,16,A6
HOLEN1          LD        A5,10,A6
                AB.L(7)   R TUR3
*
*ERROR DATAS FAULT
HOLCR3          LDK       A3,1                *UNKNOWN CHARACTER
                OR.S      A3,STATUS
                LDK       A3,/20
                RB(7)     HOLCR9
*CONVERSION TABLE
TABLE1          DATA      TABI12
                DATA      TABI11
                DATA      TABI10
TABI12          DATA      'ABCDEFGHI '
TABI11          DATA      'JKLMNOPQR '
TABI10          DATA      '/STUVWXYZ '
TABLE2          DATA      TABLE3
                DATA      TABLE4
                DATA      TABLE5
TABLE3          DATA      /5B2E               *12-2-12-3
                DATA      /3C28               *12-4-12-5
```

```
            DATA    /2B5E           *12-6-12-7
TABLE4      DATA    /2124           *11-2-11-3
            DATA    /2A29           *11-4-11-5
            DATA    /3B5D           *11-6-11-7
TABLE5      DATA    /5C2C           *10-2-10-3
            DATA    /255F           *10-4-10-5
            DATA    /3E3F           *10-6-10-7
ITCR5       EQU     *
            ANK     A2,/0F          *KEEP BITS 12,14,15
            RF(0)   ITCR51                  *STATUS = 0
            LD      A1,20,A6        *STATUS  0
            AB.L(5) ITCR2           *GO TO RETRY
            ORK.L   A2,/8000
            LDK     A1,0            *USER WANTS HARDWARE STATUS
ITCR53      LD      A5,10,A6
            AB.L(7) R TUR1
* THIS SEQUENCE PUTS CHARACTERS IN USER BUFFER IF STATUS = 0
ITCR51      LD      A5,12,A6        * USER BUFFER ADDRESS
            LD      A4,14,A6        *REQUESTED
            LDK     A1,0            *EFFECTIVE LENGHT
            LDK.L   A2,CRBUFF       *SYSTEM BUFFER ADDRESS
            CWK     A4,160
            RF(2)   *+4
            LDK     A4,160
ITCR52      LDR*    A3,A2
            STR     A3,A5
            ADK     A5,2
            ADK     A1,2
            ADK     A2,2
            SUK     A4,2
            RB(1)   ITCR52
            LDK     A2,0
            CWK     A1,160
            RF(0)   *+4
            LDK     A2,8
            RB(7)   ITCR53
*
            END
```

→:EØS

```
          IDENT    M_RETR
*THIS MODULE PROCESSES THREE FUNCTIONS
*                  * PRINT OF RETRY MESSAGE
*                  * PROCESS OF RY FUNCTION
*                  * PROCESS OF RD FUNCTION
*
          ENTRY    M_RETR
          ENTRY    RYPRO
          ENTRY    RDPRO
*
* ENTRY CONDITIONS FOR M_RETR
*                  * A6 = DWT ADDRESS
*                  * A2 = STATUS
*                  * A1 = FLAG RY OR NOT   IF 0, RY
*                  * A3 = REQUEST
* ENTRY CONDITIONS FOR RYPRO OR RDRRO
*                  * A5= INHCP ADDRESS (IMR A5)
*                  * MESSAGE ECR =ECBCP - BUFCP
*                  * IF ERROR, GO TO ERHB
*
          EXTRN    CPRTN
          EXTRN    ERHB
          EXTRN    ECBCP
          EXTRN    BUFCP
          EXTRN    HB
          EXTRN    R_TURN
          EXTRN    R_TUR4
          EXTRN    R_TUR5
*
M_RETR    ADK      A1,0
          RF(4)    RETRY1
RETRY3    LDK.L    A4,TABLE1+2 * PUT IN TABLE
          LDR*     A5,A4                *SEARCH FOR AN EMPTY WORD.
          RF(0)    RETRY2
          ADK      A4,4
          CWK      A4,TABLE2
          RB(1)    RETRY3
          RB(7)    RETRY3+4
*
RETRY2    STR      A6,A4
          SUK      A4,2
          ECR      A3,A3
          LC       A3,3,A6
          STR      A3,A4
RETRY1    LDR*     A3,A6                *PUT DN IN BUFFER
          ST       A3,BUFRY+4
          LD       A3,2,A6              *PUT DA IN BUFFER
          ANK      A3,/3F
          LDR      A4,A3
          SRL      A4,4
          ADK      A4,/30
          ECR      A4,A4
          ANK      A3,/F
          ADK      A3,/30
          CWK      A3,/3A
          RF(2)    *+4
          ADK      A3,/7
          ADR      A4,A3
          ST       A4,BUFRY+6
          LDK      A4,/30               *PUT STATUS IN BUFFER
          SC       A4,BUFRY+9
          SC       A4,BUFRY+10
          LDR      A3,A2
```

```
                SRL     A3,4
                ANK     A3,/F
                ADK     A3,/30
                SC      A3,BUFRY+11
                ANK     A2,/F
                ADK     A2,/30
                CWK     A2,/3A
                RF(2)   *+4
                ADK     A2,7
                SC      A2,BUFRY+12
                LDK     A2,15
                ADK     A1,0
                RF(4)   *+4
                ADK     A2,3
                ST      A2,ECBRY+4
                LDK     A7,6
                LDK.L   A8 ,ECBRY
                LKM
                DATA    1
                ADK     A1,0
                AB.L(0) R TURN          *RY* RETURN TO INTERRUPTED PROGRAM
                LDK     A2,0
                AB.L(7) R TUR4          *NO RY * END OF IO
TABLE1          DATA    0               *REQUEST- DA
                DATA    0               *DWT ADDRESS
                DATA    0
                DATA    0
                DATA    0
                DATA    0
                DATA    0
                DATA    0
                DATA    0
                DATA    0
                DATA    0
                DATA    0
TABLE2          DATA    0
                DATA    0
                DATA    0
BUFRY           DATA    'ℓPU,DNXX,STAT,RY'
ECBRY           DATA    /8005
                DATA    BUFRY-2
                RES     3
* THIS SEQUENCE PROCESSES RY AND RD
RYPRO           IMR     A5              *SEARCH FOR THE DEVICE ADDRESS
RDPRO           EQU     RYPRO
* GO TO HB MODULE
* A1 = CHARACTER ADDRESS
                CF      A14,HB
* RETURN A2 = BINARY RESULT
                CWK     A2,/3F
PRO1            AB.L(1) ERHB            *ERROR IN THE STATEMENT
                LDR     A3,A2           *A3 = DEVICE ADDRESS
                LDK.L   A4,TABLE1+1 * SEARCH IN TABLE
PRO6            CCR     A3,A4
                RF(0)   PRO5
                ADK     A4,4
                CWK     A4,TABLE2
                RB(1)   PRO1
                RB(7)   PRO6
*                                       * DEVICE ADDRESS FOUND
PRO5            SUK     A4,1            * IS IT RD OR RY
                LD      A1,BUFCP
                CWK     A1,/5244        *RD
                RF(0)   PRO7
                ADK.L   A3,/41C0        *RY
                LD      A6,2,A4
```

```
                LD      A5,10,A6
                LD      A1,18,A6        *SET ORDER
                ANK     A1,/FF
                SUK     A1,3
                RF(2)   PROA2
                SUK     A1,2
                RF(1)   PROA2
                LDK     A1,2
                SC      A1,19,A6
PROA2           INH
                ST      A3,EXE
                LD      A1,2,A5         * RESTORE PARAMETERS
                ST      A1,12,A6
                LDK     A1,0
                ST      A1,16,A6
                LDK     A2,0
                MS      2,24,A6
                MS      2,6,A5
                LCR     A1,A4
                ANK     A1,/F
EXE             RES     1               *CIO START
                LDK     A1,0
                STR     A1,A4           *CLEAR THE RY REQUEST
                ST      A1,2,A4
                ENB
PRO8            AB.L(7) CPRTN           *END.
*
*****************************************************************************
PRO7            LD      A6,2,A4         *RD
                LDK     A1,0
                ST      A1,2,A4         * CLEAR THE REQUEST
                STR     A1,A4
                LD      A8,10,A6
                LDK.L   A2,/8000        UPDATE STATUS
                LDK.L   A7,CPRTN
                AB.L(7) R TUR5          *UPDATE EVENT
*
                END
EOS
```

```
          IDENT    OUTPUT
          ENTRY    O TPUT
          EXTRN    E NDIO
* THIS SEQUENCE IS CALLED BY THE AND SWITCH TO SPECIFIC SEQUENCE
*                  A6= DEVICE WORK TABLE
O TPUT    ML       4,12,A6              * INITIALISE THE REGISTERS*
*
*         A1 = CHARACTER ADDRESS
*         A2 = REQUESTED LENGTH
*         A3 = EFFECTIVE LENGTH
*         A4 = ORDER
          LDK      A5,0
          SUK      A4,5
          ADR      A4,A4
          ABI(7)   SWFUNO,A4
SWFUNO    DATA     BINOUT               * BASIC OUTPUT
          DATA     ASCOUT               * ASCII OUTPUT
          DATA     OBJOT4               * STANDARD BINARY OUTPUT (4X4)
          DATA     OBJOT8               * STANDARD BINARY OUTPUT (8X8)
* THIS SEQUENCE TAKES A CHARACTER IN BUFFER AND STORES IT IN OUTWORD
* OF DEVICE WORK TABLE
*                  A6 = DEVICE WORK TABLE
BINOUT    CWR      A2,A3
          RF(0)    BINOT2
*
BINOT1    LCR      A4,A1                * IT IS NOT FINISHED
BINOT4    ADK      A1,1
          ADK      A3,1
BINOT5    MS       3,12,A6
BINOT3    ST       A4,22,A6             * PUT IN OUTWORD
          RTN      A15
*
BINOT2    AB.L(7)  E NDIO
* THIS SEQUENCE TAKES A CHARACTER AND,AT THE ENDS,OUTPUTS X-OFF,CR,LF.
*                  A6= DEVICE WORK TABLE
ASCOUT    CWR      A2,A3                * IS IT THE LAST CHARACTER
          RB(1)    BINOT1               * NO
          LDR*     A4,A6
          CWK      A4,/5450             TEST 'TP'
          RF(0)    ASRTP0
          LD       A2,26,A6             * YES
          SUK      A2,5
          RF(2)    ASC001
          LDR*     A4,A6
          CWK      A4,/5459             * 'TY'
          RB(0)    BINOT2
ASC000    EQU      *
          LDK.L    A4,/1D0A             SUK A5,10
          RF(7)    OBJ083-4
ASC001    ADK      A2,6
          LC       A4,ASCBUF,A2
          IM       26,A6
          RB(7)    BINOT3
ASRTP0    LD       A2,26,A6
          SUK      A2,10
          RB(0)    BINOT2
          ADK      A2,7                 TEST 'TAPE OFF' IN ORDER TO
          RB(0)    ASC000               INSERT SPACES BEFORE IT
          RF(2)    ASRTP1
          LD       A2,26,A6
          SUK      A2,9                 TEST
          RB(4)    ASC000               LAST SPACE
          LDK      A2,4                 A2 = POSITION OF 'TAPE OFF'
```

```
                RB(7)   ASC001+2                IN ASCBUF
ASRTP1          SUK     A2,2
                RB(7)   ASC001
ASCBUF          DATA    /A
                DATA    /130D
                DATA    /14FF
* THIS SEQUENCE TAKES A CHARACTER IN BUFFER,PERFORMS A CHECKSUM
*               A6= DEVICE WORK TABLE
OBJOT8          LCR     A4,A1
                CWK     A3,1
                RB(0)   BINOT4
                RF(2)   OBJO81
                CWR     A2,A3
                RF(0)   OBJO83-8
                XR.S    A4,24,A6
                RB(7)   BINOT4                  * GO TO STORE THE OUTPUT CHARACTER
*
                LDK.L   A4,/1D06                SUK A5,6
                ST      A4,OBJO87+2
OBJO83          LD      A4,24,A6                * LAST CHARACTER
                LD      A5,26,A6
                IM      26,A6
                ADK     A5,0
                RB(0)   BINOT3
OBJO87          LDK     A4,0
*****************************************************************************
                SUK     A5,6                    WARNING.INSTRUCTION DYNAM. MODIFIED
*****************************************************************************
                RB(4)   BINOT3
                ST      A4,24,A6
                RB(7)   BINOT2
* THIS SEQUENCE TAKES A CHARACTER IN BUFFER,PERFORMS ACHECKSUM AND
* OUTPUTS X-OFF (4X4 FORMAT)
*               A6= DEVICE WORK TABLE
OBJOT4          LCR     A4,A1
                CWK     A3,1
                RF(0)   OBJO42
                RF(2)   OEJO41
                CWR     A2,A3
                RF(0)   OBJO43
                LD      A5,26,A6
                RF(1)   OBJO44
                XR.S    A4,24,A6                * FIRST HALF CHARACTER
OBJO45          IM      26,A6
                SRL     A4,4
                RF(7)   CONVE4
*
OBJO44          LDK     A5,0                    * SECOND-HALF CHARACTER
                ST      A5,26,A6
                RF(7)   OBJO81
* THIS SEQUENCE CONVERTS THE CHARACTERS
OBJO41          ADK     A4,/8                   * FIRST CHARACTER
OBJO81          ADK     A3,1
                ADK     A1,1
CONVE4          ANK     A4,/F                   * CONVERSATION TO AVOID TROUBLIMG CHAR.
                RF(0)   CONVE5
                CWK     A4,/5
                RF(2)   CONVE6
CONVE5          ORK     A4,/10
CONVE6          RB(7)   BINOT5
*
OBJO42          LD      A5,26,A6                * SECOND CHARACTER
                RB(0)   OBJO45
                RB(1)   OBJO44
*
OBJO43          LD      A4,24,A6                * LAST CHARACTER
```

```
            LD      A5,26,A6
            RB(0)   OBJ045
            IM      26,A6
            SUK     A5,2                    * TEST IF THE SECOND PART OF CHECKSUM
*                                           * IS OUTPUT
            RB(2)   CONVE4
            RF(0)   OUTXOF
            LDK.L   A4,/1D06                SUK A5,6
            ST      A4,OBJ087+2
            CWK     A5,6
            RB(2)   OBJ087                  * OUTPUT NULL CODES
            RF(0)   OUTTOF                  * OUTPUT TAPE OFF
            CWK     A5,7
            RB(1)   BINOT2                  * END OF IO
            LDK     A4,/FF                  * OUTPUT RUB OUT
            RB(7)   CONVE6
OUTXOF      LDK     A4,/13                  * X OFF
            RB(7)   CONVE6
OUTTOF      LDK     A4,/14                  * TAPE OFF
            RB(7)   CONVE6
            END
 EOS
```

```
          IDENT     COMIO
* THIS MODULE IS COMMON TO ALL DRIVERS
          ENTRY     C INPT
          ENTRY     C OUT
          ENTRY     C WAIT
*
          EXTRN     I NPUT
          EXTRN     O TPUT
          EXTRN     R TURN
          EXTRN     PWAIT
* THIS SEQUENCE PUTS THE ASKING PROGRAM IN WAIT ON ECB
*         ENTRY CONDITIONS
*         A7 = ORDER
*         A8 = ECB ADDRESS
*         ENTRY AB.L (7)                  C  WAIT
*
C WAIT    LDR       A1,A7
          ANK       A1,/80
          AB.L(0)   R TURN
          LDK       A6,0
          AB.L(7)   PWAIT
*
*****************************************************************************
*
* THIS SEQUENCE EXECUTES AN INR INSTRUCTION
* ENTRY CONDITIONS
*         A6 = DWT ADDRESS
*         ENTRY AB.L(7)                   C INPT
*
C INPT    LD        A1,2,A6               *A1 = DEVICE ADDRESS
          ADK.L     A1,/4D00
          ST        A1,*+4                * STORE THE INR INSTRUCTION YO EXECUTE
          DATA      0                     * INPUT
          RF(4)     C OUT
          CF        A15,I NPUT
END       AB.L(7)   R TURN
*
*****************************************************************************
*
* THIS SEQUENCE EXECUTES AN OTR INSTRUCTION
* ENTRY CONDITIONS
*         A6 = DWT ADDRESS
*         ENTRY AB.L(7)                   C OUT
*
C OUT     LD        A2,22,A6              *A2 = CHARACTER TO OUTPUT
          LD        A1,2,A6
          ADK.L     A1,/4200
          ST        A1,*+4                *STORE THE OTR INSTRUCTION TO EXECUTE
          DATA      0                     * OUTPUT
          RBF (0)   COOUT1
*                   OUTPUT REFUSED
          ABI       36,A6                 BACK TO INTERRUPT SEQUENCE
COOUTI    CF        A15,0 TPUT
          RB(7)     END
*
          END
 EOS
```

```
          IDENT    INPUT
          ENTRY    I NPUT
          EXTRN    E NDIO
* THIS SEQUENCE IS CALLED BY THE DRIVERS AND SWITCH TO SPECIFIC SEQUENCE
*                  A5= CHARACTER WICH HAS BEEN INPUT
*                  A6= DEVICE WORK TABLE
I NPUT    ML       4,12,A6                 * INITIALISE THE REGISTERS
*         A1 = CHARACTER ADDRESS
*         A2 = REQUESTED LENGTH
*         A3 = EFFECTIVE LENGTH
*         A4 = ORDER
          ANK      A4,/FF
          SUK      A4,1
          ADR      A4,A4
          ABI(7)   SWFUNI,A4
SWFUNI    DATA     BININP                  * BASIC INPUT
          DATA     ASCINP                  * ASCII INPUT
          DATA     OBJIN4                  * STANDARD BINARY INPUT (X4)
          DATA     OBJIN8                  * STANDARD BINARY INPUT(8-8)
* THIS SEQUENCE STORES IN THE USER BUFFER
*                  A6 = DEVICE WORK TABLE
*                  A5= CHARACTER WICH HAS BEEN INPUT
BININP    SCR      A5,A1
          ADK      A1,1
          ADK      A3,1
          CWR      A2,A3
          RF(0)    ENDINP
BINIEN    MS       3,12,A6                 * RETURN TO CALLING PROGRAM
          RTN      A15
* THIS SEQUENCE STORES A CHARACTER AND UPDATES THE ORDER
*                  A6= DEVICE WORK TABLE
*                  A5= CHARACTER WICH HAS BEEN INPUT
ASCINP    ANK      A5,/7F                  * ANALYSE THE CHARACTER
          LD       A4,26,A6                * HAS A DELETE CODE BEEN READ
          RF(0)    ASCIN7                  * NO
          SUK      A5,/0D                  * YES
          RB(4)    BINIEN                  * IT IS NOT A CR
          LDK      A4,0                    * IT IS A CR
          ST       A4,26,A6
          RB(7)    BINIEN
ASCIN7    CWK      A5,/20                  * IS THE CHARACTER RANGING FROM 0 TO 1F
          RF(2)    ASCIN1
          CWK      A5,/5E
          RF(4)    ASCIN2
          SUR      A1,A3                   * VERTICAL ARROW
          LDK      A3,0
          IM       26,A6                   * READ UP TO CR
ASCIEN    RB(7)    BINIEN                  * RETURN TO INTERRUTED PRORAM
*
ASCIN2    CWK      A5,/5F                  * IS IT A HORIZONTAL ARROW
          RF(4)    ASCIN3
          ADK      A3,0                    * HORIZONTAL ARROW
          RB(0)    ASCIEN
          SUK      A1,1
          SUK      A3,1
          RB(7)    ASCIEN
*
ASCIN3    CWK      A5,/7F                  * IS IT A DELETE CHARACTER
          RB(0)    ASCIEN
          CWR      A2,A3
          RB(0)    ASCIEN
ASCIN6    CWK      A5,/5C                  * IS IT A TAB CHARACTER
          RF(4)    ASCI66                  * NO
```

```
                LD      A4,22,A6            * YES
                RF(0)   ASCI66              * NO TABULATION
                LCR     A7,A4               * A7 = NUMBER OF TABULATION
                ANK     A7,/FF
                RF(0)   ASCI66              * NO TABULATION
ASCI67          ADK     A4,1
                ADK     A3,1                * INCREASE TEMPORARILY EFFECTIVE LENGTH
* TACKET STARTS AT THE FIRST CHARACTER IN BUFFER
                CCR     A3,A4
                RF(2)   FILBLK              * IT IS POSSIBLE TO FILL WITH BLANK
                SUK     A3,1                RESTORE A3
                SUK     A7,1
                RF(0)   ASCI66
                RB(7)   ASCI67
FILBLK          LCR     A4,A4
                ANK     A4,/FF
                SUK     A3,1                * RESTORE A3
                SUK     A4,2
                CWR     A3,A4
                RF(0)   ASCI66
                RB(1)   ASCIEN
                LDK     A5,/20
FILBL1          SCR     A5,A1               * STORE BLANK
                ADK     A1,1
                ADK     A3,1                * UPDATE POINTERS
                CWR     A3,A2
                RB(0)   ASCIEN
                CWR     A3,A4
                RB(2)   FILBL1              * TABULATION NOT REACHED
ASCI66          SCR     A5,A1               * STORE CHARACTER
                ADK     A1,1
                ADK     A3,1
                RB(7)   ASCIEN
*
ASCIN1          ADK     A3,0                * SPECIAL CHARACTER
                RF(1)   ASCIN5
                CWK     A5,/18
                RF(2)   ASCIN4
                SUK     A5,/18
                LDK     A4,1
ASCIN8          AD.S    A4,18,A6            * UPDATA ORDER
                RB(7)   ASCI66
*
ASCIN4          CWK     A5,/10
                RF(0)   ASCIN9
                CWK     A5,/14
                RF(1)   ASCIN9
                ADK     A5,/0
                RB(0)   ASCIEN
                CWK     A5,/4
                RF(1)   ASCIN5
ASCIN9          ANK     A5,/F
                LDK     A4,2
                RB(7)   ASCIN8
ASCIN5          CWK     A5,/0D              * CARRIAGE RETURN
                RB(4)   ASCIEN
ENDINP          MS      3,12,A6
                AB.L(7) E NDIO
* THIS SEQUENCE STORES THE HALF CHARACTER AND PERFORMS CHECKSUM
*               A6 = DEVICE WORK TABLE
*               A5 = HALF CHARACTER
* THIS SEQUENCE USES OBJIN8
OBJIN4          ANK     A5,/F
                LD      A4,26,A6
                RF(0)   FIRST
                LDK     A4,0
```

```
                ST      A4,26,A6
                LCR     A4,A1
                ADR     A5,A4
                RF(7)   OBJIN8
*
FIRST           IM      26,A6
                SLL     A5,4
                SCR     A5,A1
                RB(7)   ASCIEN
* THIS SEQUENCE STORES THE CHARACTER AND PERFORMS CHECKSUM
*               A6 = DEVICE WORK TABLE
*               A5 = CHARACTER TO STORE
OBJIN8          LD      A4,10,A6
                CW      A3,4,A4
                RF(6)   OBJMOD
                SCR     A5,A1
                ADK     A1,1
OBJMOD          ADK     A3,1
                CWK     A3,2
                RF(0)   OBJI81
                XR.S    A5,24,A6        * CHECKSUM
                CWR     A3,A2
                RB(4)   ASCIEN
                LD      A5,24,A6        * TEST IF CHECKSUM IS NULL
                RB(0)   ENDINP
                LDK     A5,4
                OR.S    A5,8,A4
                RB(7)   ENDINP
OBJI81          ADR     A5,A5
                ADK     A5,3
                LDR     A7,A2
                LDR     A2,A5
                CWR     A5,A7
                RB(5)   ASCIEN
*
OBJI83          LDK     A5,8            * ERROR OF LENGTH
                OR.S    A5,8,A4
                AB.L    ASCIEN
                END
 EOS
```

```
          IDENT    INTCP
*
*
*
* INTERRUPT MODULE FOR CONTROL PANEL
* SEND CHARACTERS M  ON ASR AND ASK FOR THE COMMAND
* GO TO PROCESSOR WHEN COMMAND IS READ
*************************************************************************
*         UPON ENTRY A1,A2 ARE ALREADY IN STACK
*
          ENTRY    I ITCP
          ENTRY    INHCP
          ENTRY    INHST
          ENTRY    ERHB
          ENTRY    BUFCP,CPRTN
          ENTRY    ECBCP
          ENTRY    BH
*
*************************************************************************
          EXTRN    HB
          EXTRN    DISPAT
          EXTRN    CVTSTB
          EXTRN    CHLEV
          EXTRN    LOADER
          EXTRN    LDFLAG
          EXTRN    PCT61
          EXTRN    M CMAD
          EXTRN    RYPRO
          EXTRN    RDPRO
          EXTRN    M ASPR
          EXTRN    ABORT
          EXTRN    PAUSE
          EXTRN    RSTART
          EXTRN    MANCT
*
*************************************************************************
STADR     EQU      -4
INHCP     DATA     0                    IF AN INTERRUPT CONTROL PANEL IS
          RES      5
ZON14     DATA     0
*                                       BEING PROCESSED, ONLY HT MSG IS PERMITED
*
CSTAB1    DATA     'DM'
          DATA     DMH
CSTAB2    DATA     'HD'
          DATA     HT
          DATA     'WM'
          DATA     WM
          DATA     'LD'
          DATA     LOADER
          DATA     'ST'
          DATA     START
          DATA     'RY'
          DATA     RYPRO
          DATA     'RD'
          DATA     RDPRO
          DATA     'AS'
          DATA     M ASPR
          DATA     'AB'
          DATA     ABORT
          DATA     'PS'
          DATA     PAUSE
          DATA     'RS'
```

```
            DATA    RSTART
            DATA    'M.C'
            DATA    MANCT
            DATA    0
*
I ITCP      MSR     8,A15
            RIT     /0F
            LDK.L   A5,INHCP            A5= INHCP
            LDR*    A6,A5
            CWK     A6,2
            RF(6)   NOP
            LDK     A1,49
            CF      A15,CHLEV
* WE ARE NOT AT LEVEL 49 ,ENB, MASTER MODE
            ADK     A6,0
            RF(1)   BUSY
            ST      A14,USER14
            LDK.L   A14,ZON14
BUSY        ADK     A6,2
            STR     A6,A5
            LDK     A7,5
            LDK.L   A8,ECBCT
            LKM
            DATA    1
            LDR*    A1,A8               WAIT UNTIL OUTPUT
            RB(6)   *-2                 IS FINISHED
            LDK     A7,2
            LDK.L   A8,ECBCP            INPUT IN BUFCP
            LKM
            DATA    1
            LDR*    A1,A8
            RB(6)   *-2
            SUK     A6,2
            STR     A6,A5
            LDK     A1,0
            ST      A1,FAULT
            ST      A1,STOPFG
            LDK.L   A1,BUFCP
            LDR*    A4,A1
            ADK     A1,2
            LD      A7,ECBCP+6
            SUK     A7,2
            LDR*    A2,A5
            RF(4)   INTCP1
            LDK     A2,CSTAB1-CSTAB1
            RF      *+4
INTCP1      LDK     A2,CSTAB2-CSTAB1
INTCP2      LD      A3,CSTAB1,A2
            RF(0)   INTCP3
            CWR     A4,A3
            ABI(0)  CSTAB1+2,A2
            ADK     A2,4
            RB      INTCP2
INTCP3      IM      FAULT
            AB.L(7) ERHB
HT          IM      STOPFG
            IMR     A5
CPRTN       LDK.L   A1,FAULT
FAULT       EQU     *-2
            RF(4)   ERRET
            LD      A1,INHCP
            CWK     A1,2
            RF(0)   SECOND
MAINRT      LDK.L   A14,USER14
USER14      EQU     *-2
RETURN      LD      A1,INHCP
```

```
SECOND   EQU     RETURN
         SUK     A1,1
         ST      A1,INHCP
NOP      LDK     A6,0
         AB.L    DISPAT
ERRET    IMR     A5
         LDR*    A1,A5
         CWK     A1,2
         RB(0)   SECOND
         RB      MAINRT
**********
**********
ECBCT    DATA    5
         DATA    BUFCT
         DATA    2
         RES     3
*
**********
BUFCT    DATA    /4D3A           * 'M '
**********
ECBCP    DATA    5
         DATA    BUFCP
         DATA    72
         DATA    0
         DATA    0
         DATA    X'0020'
BUFCP    RES     40
*
*
*****************************************************************************
*****************************************************************************
*
*****************************************************************************
* WRITE MEMORY
*
* CONTENT OF BUFFER   WH ABCD VAL0 VAL1 VAL2 ---- ----
*
*
*****************************************************************************
*****************************************************************************
WM       IMR     A5
         CF      A14,HB
         CF      A14,M CMAD          GO COMPARE ADDRESS
         LDR     A3,A2               A3= ADDRESS WHERE TO WRITE
         ADK     A7,0
         AB.L(0) ERHB
WM0      ADK     A7,0
         RF(0)   WM1                 NOTHING MORE IN BUFFER
         CF      A14,HB
         STR     A2,A3               WRITE INTO MEMORY
         ADK     A3,2
         RB(7)   WM0
WM1      RB(7)   CFRTN
*
*
*
*****************************************************************************
*****************************************************************************
*
* HEXADECIMAL MEMORY DUMP
*        THIS MODULE OUTPUT THE CONTENT OF CORE MEMORY
*        ON ASR , BETWEEN TWO GIVEN ADDRESS
*
*        SYNTAX OF THE COMMAND LINE   DH VAL1 VAL2
*        WHERE VAL1 AND VAL2 ARE TWO HEXADECIMAL ADDRESSES
*
```

```
*
*         USE OF REGISTERS   A1 THRU A7
*
*         DMH USES A1,A2,A3,A4,A5,A6,A7
*         BH  USES A4,A5,A6,A7,A1
*
*
********* VAL1,VAL2 ARE ALREADY IN BUFCP
*
*
*
*****************************************************************************
DMH       IMR       A5
          CF        A14,HB
          CF        A14,M CMAD              GO COMPARE ADDRESS
          LDR       A3,A2                   A3 = FIRST ADDRESS
          ANK.L     A3,/FFF0
          CF        A14,HB
          CF        A14,M CMAD              GO COMPARE ADDRESS
          ORK       A2,/E
          ADK       A2,2
          SRL       A2,1
          SRL       A3,1
          CWR       A3,A2
          AB.L(1)   ERHB
          SLL       A2,1
          SLL       A3,1
          LDK       A1,/31
          ST        A1,BUFCP-2
          LDK       A6,0
          CF        A14,CIO
          LDK       A1,/20
          ST        A1,BUFCP-2
DMH0      LDK.L     A6,BUFCP                A6= ADDRESS WHERE TO STORE CHARACTERS
          LDK.L     A1,/2020
          LDR       A7,A3
          CF        A14,BH                  BH CONVERTS NUMBER A7 AND STORES
*                                           IT IN BUFCP
          STR       A1,A6                   STORE TWO MORE BLANKS
          ADK       A6,2
*
*
          LDK       A5,0                    COUNT OF WORDS
          LDR*      A7,A3                   TAKE FIRST WORD
*
DMH1      ADK       A5,1
          ADK       A3,2
          CWR       A3,A2                   IT IS FINISH
          RF(0)     DMHA
          LDR*      A4,A3                   A4 = NEXT WORD
          CWR       A4,A7                   IS IT THE SAME THAN FIRST WORD
          RB(0)     DMH1                    YES,LOOP
*
          CWK       A5,7                    NO ARE WE STILL ON FIRST LINE
          RF(1)     DMHA                    NO GO TO PRINT BLANKS
          LDK       A4,0                    USE, PREPARE THE LINE
DMH2      CF        A14,BH                  CONVERT AND STORE THE FIRST WORD
          CF        A14,STOINT STOREINTERPRETATION
          ADK       A4,1
          CWR       A5,A4
          RB(1)     DMH2
*
DMH3      LDR*      A7,A3
          ADK       A3,2
          CF        A14,BH                  COVERT AND STORE THE
          CF        A14,STOINT              REMAINDER OF THE LINE
```

```
                ADK       A4,1
                CWK       A4,8
                RB(2)     DMH3
                LDK       A6,72
DMH5            CF        A14,CIO             CALL OUTPUT FUNCTION
                CWR       A3,A2
                RB(4)     DMH0                NO, LOOP TO BEGIN
                AB.L(7)   CPRTN               RETURN
ECBDH           DATA      2                   LISTING STANDARD
                DATA      BUFCP-2
                DATA      73
                RES       3
* THIS IS THE CASE WHERE ALL THE NUMBERS OF THE LINE
*                 ARE DENTICAL
DMHA            CF        A14,BH              CONVERT AND STORE FIRST WORD
                ANK.L     A3,/FFF0
                LDK       A6,12               NUMBER OF CHARACTER IN BUFCP
                RB(7)     DMH5
*
*
*****************************************************************************
*****************************************************************************
*
*
CIO             LDK.L     A7,STOPFG
STOPFG          EQU       *-2
                RF(0)     *+6
                AB.L      CPRTN
                ADK       A6,2
                ST        A6,ECBDH+4
                LDK.L     A8,ECBDH
                LDK       A7,6
                LKM
                DATA      1
                LDR*      A7,A8
                RB(6)     *-2
                RTN       A14
*
*****************************************************************************
*****************************************************************************
* STORE INTERPRETATION ROUTINE
* A7 CONTAINS THE CHARACTERS TO BE STORED
*
*
STOINT          STR       A2,A14
                SUK.L     A14,2
                STR       A7,A14
                SLL       A4,1
                LDK       A2,0
                ECR       A8,A7
DMH2C           ANK                           A7,/FF
                CWK       A7,/20
                RF(2)     DMH2A
                CWK       A7,/60
                RF(2)     DMH2B
DMH2A           LDK       A7,/20
DMH2B           ADK       A2,0
                RF(1)     DMH2D
                SC        A7,BUFCP+57,A4
                LDR       A7,A8
                ADK       A2,1
                RB(7)     DMH2C
DMH2D           SC        A7,BUFCP+56,A4
                SRL       A4,1
                LDR*      A7,A14
                ADK.L     A14,2
```

```
            LDR*    A2,A14
            RTN     A14
*
*
*****************************************************************************
*****************************************************************************
*
*****************************************************************************
*****************************************************************************
* BH SUBROUTINE CONVERT THE NUMBER GIVEN IN A7
*       INTO    HEXADECIMAL CHARACTERS , AND STORE IN BUFCP
*       AFTER THAT PUT A BLANK LOCATION IN BUFCP
*
*
*       UPON ENTRY                      A7= NUMBER (BINARY) TO CONVERT
*                                       A6= ADDRESS OF STORING AREA
*                                       A1= BLANKS
*
*       UPON EXIT                       A6= UPDATED
*                                       A7 NOT DESTROYED
*
*
*****************************************************************************
*****************************************************************************
BH      STR     A5,A14
        SUK.L   A14,2
        STR     A4,A14
        LDK     A4,4
BH0     SRC     A7,12
        LDK     A5,X'F'                 CONVERT
        ANR     A5,A7
        ADK     A5,X'30'
        CWK     A5,X'3A'
        RF(2)   BH2
        ADK     A5,X'7'
BH2     SCR     A5,A6                   STORE VIA A6
        ADK     A6,1
        SUK     A4,1
        CWK     A4,0                    IS IT FINISHED
        RB(1)   BH0
        SCR     A1,A6
        ADK     A6,1
        SCR     A1,A6
        ADK     A6,1
        LDR*    A4,A14
        ADK.L   A14,2
        LDR*    A5,A14
        RTN     A14
*****************************************************************************
*****************************************************************************
*****************************************************************************
START   IMR     A5
        LDK.L   A1,INHST
INHST   EQU     *-2
        RF(1)   ERHB
        LD      A1,LDFLAG
        RF(0)   ERHB
        LD      A1,PCT61+STADR
        RF(0)   ERHB
        IM      INHST
        AB.L    CPRTN
*
*****************************************************************************
*****************************************************************************
*
ERHB    LDK.L   A8,ECBER
```

```
            LDK     A7,6
            LKM
            DATA    1
            LDR*    A1,A8
            RB(6)   *-2
            AB.L    CPRTN
*
*****************************************************************************
ECBER       DATA    5
            DATA    ERMSG-2
            DATA    4
            RES     3
            DATA    0
*
ERMSG       DATA    'ER'
*
            END
 EOS
```

```
                IDENT   ABORT
* THIS MODULE PROCESS THE ABORT CALL , FROM ANY
*               INTERRUPT OR MACRO PROCESSOR
*
* AND THE 'BRANCH ON LABEL IN CASE OF ABORT' MACRO
*
                ENTRY   ABADR
                ENTRY   SYSAB
                ENTRY   ABORT
                ENTRY   MCABFL
                EXTRN   DISPAT
                EXTRN   RINIT
                EXTRN   BH
                EXTRN   CHLEV
**              ABORT   PROCESSING
*
*               IF A MACRO ABADR HAS BEEN SEND, GIVE CONTROL TO
*               USER VIA PARAMETERS BLOCK.
*
*               IF NOT SEND MESSAGE ABORT CODE ADDR
*
*
*               IF ABORT FROM OPERATOR , BRANCH TO RINIT
                RES     4
ZON14           EQU     *-2
ECBAB           DATA    5
                DATA    ABBUF
                DATA    16
                RES     2
ABBUF           DATA    /0030
                DATA    'AB  '
CODE            DATA    'CD  '
ADDR            DATA    'ADDR'
                DATA    0
*  MACRO ABADR  PROCESSING
*
*               UPON ENTRY                      A8=LABEL
*                                               A7=ADR OF A 3-WORD BLOCK WICH WILL CONTAI
*                                               ABORT PARAM  1ST WORD  ABORT CODE
*                                                            2ND WORD  ABORTED PSW
*                                                            3TH WORD  ABORTED ADDRESS
*
*       ' *** WARNING  THIS SEQUENCE IS NOT A SCHEDULED LABEL SEQUENCE
*
*
ABADR           ST      A7,CTRLBL               CBA
                ST      A8,CTRLBL+2             LABEL
                IM      MCABFL
                LDK     A7,0
                ST      A7,4,A15                NORMAL RETURN
                AB.L    DISPAT
CTRLBL          DATA    0
                DATA    0
SYSAB           LDK.L   A1,MCABFL
MCABFL          EQU     *-2
                RF(0)   ABRT1
                LD      A7,CTRLBL               TAKE CONTROL BLOCK ADDRESS
                STR     A2,A7                   STORE CODE IN BLOCK+0
                LD      A2,18,A15
                ST      A2,2,A7                 STORE USER PSW IN BLOCK+2
                ST      A3,4,A7                 STORE ABORTED ADDRESS IN BLOCK+4
                LD      A7,CTRLBL+2             TAKE USER ABORT ADDRESS
                ST      A7,20,A15
```

```
                LDK       A6,0                NO SCHEDULED LABEL
                ST        A6,MCABFL
                AB.L      DISPAT
*
*
ABRT1           ST        A14,SAV14
                LDK.L     A14,ZON14
                LDK       A1,48
                CF        A15,CHLEV           LEVEL 48,END, C. PANEL INTERRUPT
*                                             CANT GO THRU
                LDK       A1,/20
                ADK.L     A2,/2030
                ST        A2,CODE             PUT MESSAGE IN
                LDR       A7,A3               BUFFER
                LDK.L     A6,ADDR
                CF        A14,BH
*
*
                LDK.L     A8,ECBAB            OUTPUT MESSAGE
                LDK       A7,6
                LKM
                DATA      1
*
                LDR*      A1,A8
                RB(6)     *-2
*
                LDK.L     A14,SAV14
SAV14           EQU       *-2
*
ABORT           MLR       8,A15
                AB.L      RINIT
                END
 EOS
```

```
                IDENT   PAUSE
                ENTRY   PAUSE
                ENTRY   PSMAC
                EXTRN   CPRTN
                EXTRN   INHST
                EXTRN   ERHB
                EXTRN   DISPAT
                EXTRN   PCT61
                EXTRN   CHLEV
STATUS          EQU     0
PAUSE           IMR     A5
                LD      A1,INHST
                AB.L(0) ERHB
PAUSE1          LDK.L   A1,/4000
                OR.S    A1,PCT61+STATUS
                AB.L    CPRTN
*
*
*
PSMAC           LDK     A1,48
                CF      A15,CHLEV
                LDR     A2,A7
                LDR     A1,A8
                MS      2,ECBMES+2
                LDK.L   A8,ECBMES
                LDK     A7,6
                LKM
                DATA    1
                LDR*    A1,A8
                RB(6)   *-2
                LDK.L   A1,/4000
                OR.S    A1,PCT61+STATUS
                AB.L    DISPAT
*
*
*
*
ECBMES          DATA    5
                DATA    0
                DATA    0
                RES     2
                DATA    0
*
                END
 EOS
```

```
                IDENT   MANCT
                ENTRY   MANCT
                EXTRN   ERHB
MANCT           IMR     A5
                AB.L    ERHB
                END
 EOS
```

```
          IDENT    LOADER
*
*********************************************************************************
* SYSTEM LOADER
*
*         THIS MODULE LOADS A USER PROGRAM AT BASE ADDRESS CVTBKA+16
*         HAVING FIRST LEFT 16 WORDS FOR THE USER SAVE AREA
*
*         SYNTAX OF THE COMMAND LINE     LD M
*                                        WHERE M IS OPTIONAL(MASTER MODE FLAG)
*                   USE OF REGISTERS  A1 THRU A8
*
*
*********************************************************************************
*
          ENTRY    LOADER,LDFLAG
*
          EXTRN    PCT61
          EXTRN    CVTMSZ
          EXTRN    CVTBKA
          EXTRN    CVTSBA
          EXTRN    CVTBBA
          EXTRN    CPRTN
          EXTRN    BUFCP
          EXTRN    INHCP
          EXTRN    ERHB
          EXTRN    INHST
          EXTRN    BH
          EXTRN    HB
*
          EXTRN    ECBCP
USPSW     EQU      /F800
STADR     EQU      -4
SAVADR    EQU      -2
COMAR     EQU      96                  48 WORDS COMMUNICATION AREA
*
LDFLAG    DATA     0
*
ECBCLU    DATA     4
          DATA     BUFCP
          DATA     70
          RES      2
          DATA     0
*
*
LOADER    IMR      A5
          IMR      A5
          LD       A2,INHST
          RF(1)    ER
          LDK      A2,1
          ST       A2,MASTER
          LD       A9,CVTBKA
          ADK.L    A9,COMAR
          ADK      A7,0
          RF(0)    PRAFL
          CWK      A7,2
          RF(4)    LOAD2
LOAD1     LCR      A2,A1
          CWK      A2,/20
          RF(4)    ER
          ADK      A1,1
          LCR      A2,A1
          CWK      A2,/4D              4D=M
```

```
                RF(4)   ER
                LDK     A1,0
                ST      A1,MASTER
                RF      PRAFL
        *
        *
        ER      AB.L    ERHB
        *
        LOAD2   CF      A14,HB
                ADR     A9,A2
                LDK     A2,0
                SUK     A7,0
                RB(1)   LOAD1
        *
        PRAFL   ST      A9,PCT61+SAVADR
                ADK.L   A9,32           SKIP SAVE AREA
                LD      A10,CVTMSZ      A10=ENDADDRESS
                CF      A14,TESTOV
                LDR     A11,A9
        *
        RAFL    LDK.L   A1,0
        EOTFL   EQU     *-2
                RF(0)   RAFL1
                LDK.L   A1,/2045        * E
                LDK.L   A2,/4F54        * OT
                CF      A14,HLTASC
                HLT                     HALT, CHANGE TAPE AND RESTART
                LDK     A1,0
                ST      A1,EOTFL
        RAFL1   LDK     A7,2
                LDK     A1,70
                ST      A1,ECBCLU+4
                LDK     A5,4
                CF      A14,STIO
        *
                LD      A1,ECBCLU+8
                ANK     A1,/1F
                SRC     A1,1
                RF(2)   EOF
                SRC     A1,2
                RF(2)   CLCO1           ERRONEOUS CLUSTER
                SRC     A1,1
                RF(2)   CLCO1
                SRC     A1,1
                RF(6)   LOAD3
                IM      EOTFL           END OT TAPE
        *
        LOAD3   LDK     A1,0
                LC      A1,BUFCP
                CWK     A1,/20          IS IT A CLUSTER
                RF(2)   PROLO1+2        YES GO TO PROCESS IT
        *
                LDK     A1,/20
                LDR     A7,A11
                LDK.L   A6,BUFCP
                AD      A6,ECBCLU+6
                ADK     A6,2
                CF      A14,BH
                CF      A14,ASCII       GO TO PRINT IDENT
        *
                RB      RAFL            BACK TO READING
        ********************************************************************************
        EOF     LDK     A7,/22          OUTPUT EOF
                CF      A14,ASCII+2
                LD      A1,PCT61+STADR
                RF(4)   MASTER-2
```

```
                LDK.L   A1,/204E            204E= N
                LDK.L   A2,/5320            5320=S
                CF      A14,HLTASC
                RF      EXIT
                LDK.L   A1,MASTER
MASTER          EQU     *-2
                AD.S    A1,PCT61+STADR
EXIT            IM      LDFLAG
EXIT1           LDK.L   A1,-1
                AD.S    A1,INHCP
                AB.L    CFRTN
*****************************************************************************
*
HLTASC          ST      A1,BUFCP
                ST      A2,BUFCP+2
                LDK     A1,6
                ST      A1,ECBCLU+4
ASCII           LDK     A7,6
                LDK     A5,5
                LDK.L   A1,-2
                AD.S    A1,ECBCLU+2
STIO            LDK.L   A8,ECBCLU
                STR     A5,A8
                LKM
                DATA    1
                LDR*    A2,A8
                RB(6)   *-2
                LDK.L   A1,BUFCP
                ST      A1,ECBCLU+2
                LDK     A1,72
                ST      A1,ECBCLU+4
                RTN     A14
*****************************************************************************
* PROCESS LOADING   THIS MODULE READ ACLUSTER
*       AND BRANCH ACCORDING TO THE CLUSTER TYPE
*
*       ON EXIT  A1= BUFF ADDRESS PLUS ONE
*                A2= WORD COUNT
*                A3= TYPE
*       THE TYPE MUST BE   3,4,7   IF NOT THIS   HALT
*****************************************************************************
ABA             EQU     0
PROLO1          RB(7)   RAFL                READ A CLUSTER
PROLO           EQU     PROLO1
                LDR     A9,A11
                LDK.L   A1,BUFCP
                LDK     A2,0
                LDK     A3,0
                LDK     A4,1
                LCR     A3,A1               A3 = TYPE
                ADK     A1,1
                LCR     A2,A1               A2 = WORD COUNT
                ADK     A1,1
                CWK     A3,3
                RF(0)   CLCODE              BRANCH ON CLUSTER CODE
                CWK     A3,4
                RF(0)   CLIMOD              INTERNAL MODIFICATION
                CWK     A3,7
                RF(0)   CLEND               END/START
                RB(7)   PROLO1
*
CLCO1           LDK.L   A1,/2045            2045=E
                LDK.L   A2,/4320            4320=C
                CF      A14,HLTASC
                RB      EXIT1
*
```

```
*
*
*****************************************************************************
*****************************************************************************
* CLUSTER CODE (TYPE 3)
*         UPON ENTRY    A1=ADDRESS OF BUFF+1 (RBK)  A4=1
*                       A2= WORD COUNT
*                       A9= BADDRESS
*                       A10=ENDADDRESS
*****************************************************************************
*
*
CLCODE    LD        A3,BUFCP+6
          RB(4)     PROLO1              EMBK SET SKIP THE CLUSTER
CLCO1A    LD        A3,BUFCP+4
          TM        A3,A4               IS IT RELOCATABLE SECTION
          RF(0)     CLCO2               NO
          ADR       A3,A9               YES
          CF        A14,TESTAD
          RF(7)     CLCO4
CLCO2     LDK.L     A9,ABA
CLCO4     LDR*      A5,A1               A5=(RBK)
          ADK       A1,6                A1=ADDRESS OF FIRST CODE WORD IN BUFFER
          SUK       A2,3                A2= NUMBER OF CODE WORD
*                                       A3= STORAGE ADDRESS
*                                       A4= MASK FOR RBK
*                                       A6= CODE WORD
CLCO5     SRC1      A4
          LDR*      A6,A1
          TM        A5,A4
          RF(0)     CLCO7
          ADR       A6,A11
CLCO7     STR       A6,A3               YES STORE CODE WORDS
          ADK       A1,2                UPDATE
          ADK       A3,2
*                                       POINTERS
          SUK       A2,1
          RB(4)     CLCO5
          RB(7)     PROLO
*
*
*
*****************************************************************************
* INTERNAL MODIFICATION CLUSTERS
*
*         UPON ENTRY    A1 = ADDRESS OF BUFFER+1 (RBK)
*                       A2 = WORD COUNT
*                       A9 = BASE ADDRESS
*                       A10= END ADDRESS
*****************************************************************************
CLIMOD    LDR*      A5,A1               A5= (RBK)
          SUK       A2,1
CLIM1     LDK       A7,1                A7= MASK FOR ADDRESS
          SRC       A4,1
          ADK       A1,2
          LDR*      A3,A1               A3= ADDRESS
          TM        A3,A7               IS IT RELOCATABLE
          RF(0)     CLIM2               NO
          ADR       A3,A9               YES ADD BASE
          CF        A14,TESTAD
CLIM2     ADK       A1,2                YES
          LDR*      A6,A1               TAKE CODE WORD
          TM        A5,A4               IS IT RELOCATABLE
          RF(0)     CLIM3               NO
          ADR       A6,A11
CLIM3     STR       A6,A3               YES STORE CODE WORD
```

```
*                                                   UPDATE
            SUK     A2,2                            POINTERS
            RB(4)   CLIM1                           CONTINUE
            RB(7)   PROLO
*
*
*
*****************************************************************************
*****************************************************************************
* CLUSTER           END/START
*         UPON ENTRY    A1 = ADDRESS OF BUFF+1 (START ADDRESSE
*                       A2 = WORD COUNT
*                       A9 = BADDRESS
*                       A10= ENDADDRESS
CLEND       LDR*    A3,A1
            RF(0)   CLEN3A                          FINISHED NO START ADDRESS
            TM      A3,A4                           IS START ADDRESS RELOCATABLE
            RF(0)   CLEN1                           NO
            ADR     A3,A9                           YES
CLEN1       ANK.L   A3,/FFFE
            CF      A14,TESTAD
            ST      A3,PCT61+STADR
CLEN3A      LD      A1,BUFCP+6                      UPDATE BASE ADDRESS
            ADR     A9,A1
            LDR     A11,A9
            CF      A14,TESTOV
            LDK     A1,0
            ST      A9,CVTSBA
            STR     A1,A9                           INITIALIZE GET CORE AREA
            LD      A1,CVTMSZ
            SUK     A1,2
            ST      A1,CVTBBA
            RB      PROLO
*
*
*****************************************************************************
*
OUTOV       LDK.L   A1,/204F                        204F=0
            LDK.L   A2,/5620                        5620==
            CF      A14,HLTASC
*****************************************************************************
            AB.L    EXIT1
*****************************************************************************
*
*****************************************************************************
*
TESTAD      LDR     A7,A1
            LDR     A8,A2
            LDR     A1,A9
            LDR     A2,A10
            SRL     A1,1
            SRL     A2,1
            SRL     A3,1
            CWR     A3,A1
            RB(2)   OUTOV
            CWR     A3,A2
            RB(1)   OUTOV
            SLL     A3,1
            LDR     A1,A7
            LDR     A2,A8
            RTN     A14
*
*****************************************************************************
*
TESTOV      LDR     A1,A9
            LDR     A2,A10
```

```
                RF(0)     TSTOV1
                SRL       A1,1
                SRL       A2,1
                CWR       A1,A2
                RB(1)     OUTOV
                RTN       A14
TSTOV1          ADK       A1,0
                RB(6)     OUTOV
                RTN       A14
*
*****************************************************************************
                END
```

X → EOFS

```
          IDENT     M_CMAD
* THIS SUBROUTINE COMPARE THE ADDRESS CONTAINED IN A2
*                  WITH CVTMSZ MEMORY SIZE
*                  THIS ROUTINE DESTROYS A4
*
*********************************************************************************
          ENTRY     M_CMAD
          EXTRN     CVTMSZ,ERHB
*
M_CMAD    LD        A4,CVTMSZ
          RF(0)     CMADEX              CVTMSZ NULL = 32K
          SRL       A4,1
          SRL       A2,1
          CWR       A2,A4
          AB.L(6)   ERHB
          SLL       A2,1
CMADEX    RTN       A14
          END
 EOS
```

```
                IDENT     HEBIN
*
*****************************************************************************
* THIS MODULE CONVERTS A NUMBER IN HEXADECIMAL
*       FORM INTO A BINARY FORM
*       UPON ENTRY , THE NUMBER IS IN BUFF, CHARACTER ADDRESS IN A1
*       EXIT IF BLANK OR END OF BUFF ENCOUNTERED
*       A7 GIVE THE CHARACTERS NUMBER STILL IN BUFFER.
*       A2 BINARY RESULT
*****************************************************************************
*       A4,A5,A6 CAN BE USED AS WORK REGISTERS
                ENTRY     HB
                EXTRN     ERHB
*
HB      ADK       A1,1
        SUK       A7,1
        RF(0)     ER3
HB      LDK       A2,0
        LDK       A4,0
        LCR       A4,A1
        CWK       A4,X'20'
        RB(0)     HB0
AGAIN   CWK       A4,X'30'              30=0
        RF(2)     ER3
        CWK       A4,X'3A'
        RF(2)     CHIF
        CWK       A4,X'41'              41=A
        RF(2)     ER3
        CWK       A4,X'46'
        RF(1)     ER3
        ADK       A4,9
CHIF    ANK       A4,X'F'
        SLL       A2,4
        ORR       A2,A4
        ADK       A1,1
        SUK       A7,1
        RF(0)     HB2
        LCR       A4,A1
        CWK       A4,X'20'
        RF(0)     HB2
        CWK       A4,X'2E'
        RB(4)     AGAIN
HB2     RTN       A14                   NORMAL EXIT
ER3     AB.L(7)   ERHB                  ERROR EXIT
*
*
*****************************************************************************
                END
 EOS
```

```
          IDENT    CHLEV
* THIS ROUTINE INSERT THE CALLING
* PROGRAM IN THE A15-STACK ACCORDING TO THE LEVEL
* SPECIFIED IN A1
* REGISTERS A5,A6 ARE SAVED IN THE STACK
* REGISTERS A1 TO A4 DESTROYED
*
*         CALLING SEQUENCE              A1= LEVEL (NORMALLY 48-49)
*
*                                       CF  A15,CHLFV
*
*         AT THE END OT THE ROUTINE ,BRANCH TO THE DISPATCHER
*
          ENTRY    CHLEV
          EXTRN    DISPAT
*
*
SAVE      RES      8
SAV15     DATA     0
CHLEV     MS       8,SAVE
          LDR      A3,A15
          ADK      A3,2
*
CHLEV1    ADK      A3,20                FIND AN ENTRY SO
*                                       THAT LEVEL OF THIS ENTRY =A1
          LDR*     A2,A3
          SRL      A2,10
          CWR      A2,A1
          RB(2)    CHLEV1
*
          SUK      A3,18
*
          LDR      A2,A15               SHIFT UP THE STACK OF 10 LOCATIONS
CHLEV2    ADK      A2,2                 FROM THE TOP END TO THE ENTRY
          LDR*     A4,A2
          ST       A4,-20,A2
          CWR      A2,A3
          RB(4)    CHLEV2
          SUK.L    A15,20
*                                       SET ENTRY ELEMENT
          LD       A2,4,A15
          STR      A2,A3                AO SETTING
*
          LD       A2,2,A15             PSW SETTING
          ANK.L    A2,73FF
          SLL      A1,10
          ORR      A1,A2
          ST       A1,-2,A3
          ST       A15,SAV15
          SUK      A3,4
          LDR      A15,A3
          ML       8,SAVE
          MSR      8,A15
          LD       A15,SAV15
*
          ADK.L    A15,4
*
          LDK      A6,0                 NO SCHEDULED LABEL PLEASE
          AB.L     DISPAT
          END
 EOS
```

```
          IDENT    ENDIO
* THIS MODULE CONTAINS ALL THE OUTPUTS OF THE MODULES
          ENTRY    E NDIO
          ENTRY    E S015
          ENTRY    E S014
          ENTRY    E S013
          ENTRY    E S012
          ENTRY    E S011
          ENTRY    E S000
          ENTRY    L VCH
          ENTRY    R TURN
          ENTRY    R TUR1
          ENTRY    R TUR2
          ENTRY    R TUR3
          ENTRY    R TUR4
          ENTRY    R TUR5
          EXTRN    DISPAT
          EXTRN    PCT61
* THIS SEQUENCE RETURNS TO THE CALLING SEQUENCE WITH LEVEL 48
* ENTRY CONDITIONS
*         A1 = RETURN ADDRESS
*         ENTRY    AB.L(7)              L VCH
*
L VCH     INH
          STR      A1,A15
          LDK.L    A1,/C000
          STR      A1,A15
          RTN      A15
*************************************************************************
*
* THIS SEQUENCE EXECUTES THE CIO HALT FOR PAPER EQUIPMENT
* ENTRY CONDITIONS
*         A6 = OWT ADDRESS
*         AB.L(7)  E NDIO
*
E NDIO    LD       A1,2,A6              * A1 = DEVICE ADDRESS
          ADK.L    A1,/4280
          ST       A1,*+4
          CIO      A2,0,0               * STOP DEVICE
          ADK.L    A15,4
*
R TURN    INH                           * RETURN TO INTERRUPTED PROGRAM
          MLR      8,A15
          RTN      A15
*
*************************************************************************
*
*
* END OF IO
*         A2 = STATUS
*
R TUR4    LD       A1,16,A6             * A1 = EFFECTIVE LENGTH
          LD       A5,10,A6             * A5 = ECB ADDRESS
          RF(7)    R TUR1
*
ERSOFT    LDK      A1,0
          RF(7)    *+4
*
R TUR1    LDR      A8,A5                * A8 = ECB ADDRESS
          LDK.L    A7,DISPAT
*
R TUR5    LDK      A3,/80               * A3 = EVENT BYTE
          SCR      A3,A8
```

```
            MS      2,6,A8
            CWK     A2,/C001
            RF(0)   RETU11
            SC*     A3,32,A6          * CONTROLLER STATUS
            LD      A6,30,A6
RETU11      INH
            LD      A1,PCT61
            ANK     A1,/7F
            RF(0)   RETU12
            LDK.L   A1,-1
            AD.S    A1,PCT61
RETU13      LDR     0,A7
RETU12      LDK     A6,0
            LDR     0,A7
*
* THIS SEQUENCE GIVE STATUS FOR ERRORS
E SO00      LDK     A2,0
            LDK     A1,0
            MS      2,6,A8
            LDK     A3,/80
            SCR     A3,A8
            LDK.L   A7,DISPAT
            RB      RETU11
E SO11      LDK.L   A2,/C010          * FUNCTION
            RB(7)   ERSOFT
E SO12      LDK.L   A2,/C008          * BUFFER SIZE ADDRESS
            RB(7)   ERSOFT
E SO13      LDK.L   A2,/C004          * ECB ADDRESS
            RB(7)   ERSOFT
E SO14      LDK.L   A2,/C002          * DEVICE ATTACHED
            RB(7)   ERSOFT
E SO15      LDK.L   A2,/C001          * FILE CODE
            RB(7)   ERSOFT
*
* END OF IO * SOFTWARE
R TUR2      LD      A5,10,A6          * A5 = ECB ADDRESS
AGAIN       LD      A2,8,A5
            RB(4)   R TUR4            * SOFTWARE ERROR IN READING
*
R TUR3      LDK.L   A1,*+8            * CHANGE LEVEL
            AB.L(7) L VCH
            SUR     A8,A8
            LD      A1,18,A6          * IS IT ORDER 2
            ANK     A1,/FD
            RF(4)   SUITE
            ML      3,12,A6           * IS IT AN EOS OR EOF
            CWK     A3,4              * IS THER FOUR CHARACTER
            RF(4)   FILL
            SUR     A1,A3
            LDR*    A4,A1
            CWK     A4,/3A45          *  E
            RF(4)   FILL
            LD      A4,2,A1
            CWK     A4,/4F46          * OF
            RF(4)   FILL3
            LDK     A4,/01
            RF(7)   FILL4
FILL3       CWK     A4,/4F53          * OS
            RF(4)   FILL
            LDK     A4,/02
FILL4       LDR     A8,A4
FILL        LD      A1,12,A6
            LDK.L   A4,/2020          * FILL THE BUFFER WITH BLANK
            SUR     A2,A3
            RF(0)   SUITE
            SCR     A4,A1
```

```
            ADK     A1,1
            SUK     A2,1
            RF(0)   SUITE
FILL1       STR     A4,A1
            ADK     A1,2
            SUK     A2,2
            RB(1)   FILL1
SUITE       LD      A1,18,A6
            RF(2)   AGAIN1
END3        LDR     A2,A8
            LD      A1,16,A6
            RB(7)   R TUR1
AGAIN1      LDR     A1,A8
            ANK     A1,/03
            ECR     A1,A1
            AN      A1,18,A6
            RB(4)   END3
AGAIN3      LDK     A4,2            * ORDER 2
            SC      A4,19,A6
AGAIN4      ML      2,2,A5          * READ AGAIN
            LDK     A3,0
            MS      3,12,A6
            LDK     A1,0
            LDK     A2,0
            MS      2,24,A6
            MS      2,6,A5
            AB.I(7) 6,A6
            END
 EOS
```

```
            IDENT     M ASPR
            ENTRY     M ASPR
            EXTRN     CPRTN
            EXTRN     HB
            EXTRN     ERHB
            EXTRN     P DWLG          LENGTH OF ONE DWT BLOCK
            EXTRN     D WT            DEVICE WORK TABLE
            EXTRN     F CT            FILE CODE TABLE
            EXTRN     D WTEN          END OT DWT
FCTY        EQU       5
M ASPR      IMR       A5
            CF        A14,HB
            LDR       A3,A2           A3= FCODE
            CWK       A3,FCTY
            AB.L(0)   ERHB            ASSIGN ON THE TY FILE CODE FORBIDDEN
            SLL       A3,1            A3= FCT INDEX
            CW        A3,F CT         NOT REFERENCED FILE CODE
ERROR       AB.L(1)   ERHB
            ADK       A1,1
            LCR       A4,A1
            SLL       A4,8
            ADK       A1,1
            LCR       A4,A1           A4 = DEVICE NAME
            ADK       A1,1
            CWK       A4,/4E4F
            RF(4)     ASPR1A
            LDK       A5,0
            RF        ASPR2A
*
ASPR1A      SUK       A7,3
            LDR       A6,A4
            CF        A14,HB
*                                     A2 = XX DEVICE ADDRESS
            LDK.L     A5,D WT
ASPR01      CWR*      A6,A5
            RF(4)     ASPR02          NAME NOT FOUND
            CW        A2,2,A5
            RF(4)     ASPR02          ADDRESS NOT FOUND
ASPR2A      ST        A5,F CT,A3      PERFORM THE ASSIGN
            AB.L      CPRTN
*
ASPR02      ADK.L     A5,P DWLG
            CWK       A5,D WTEN
            RB(2)     ASPR01          LOOP
            AB.L      ERHB            THE NAME DOES NOT EXIST IN TABLE
            END
 EOS

 EOF
```