```
VASM ROM ASSEMBLY          REV.  6/81A            HP-82160A HP-IL MODULE

OPTIONS: L C S          HP-IL CAS&CTL             ADDRESSES @70000-71777

     2                     FILE   SCPL1B           ILCAS&CTL QUAD 0 = CS0
     3    0          34 CON    28                 ROM ID 0028
     4    1          52 CON    42                 # OF FUNCTIONS
     5    2           0 DEFR4K CASSET       00 - MODULE ID: -MASS ST 1H
     5    3           0
     6    4           0 DEFR4K CREATF       01 - CREATE A CASSETTE FILE
     6    5           0
     7    6           0 DEFR4K DIR          02 - DISPLAY TAPE DIRECTORY
     7    7           0
     8   10           0 DEFR4K NEWTAP       03 - FORMAT TAPE BEFORE USE
     8   11           0
     9   12           0 DEFR4K PURGEF       04 - PURGE A FILE FROM TAPE
     9   13           0
    10   14           0 DEFR4K READA        05 - READ ALL INFO FRM FILE
    10   15           0
    11   16           0 DEFR4K READK        06 - READ "KEY" ASSIGN FILE
    11   17           0
    12   20           0 DEFR4K READP        07 - READ PROGRAM FROM TAPE
    12   21           0
    13   22           0 DEFR4K READR        08 - READ REGISTERS FM TAPE
    13   23           0
    14   24           0 DEFR4K READRX       09 - READ BLOCK USING X-REG
    14   25           0
    15   26           0 DEFR4K READS        10 - READ CALCULATOR STATUS
    15   27           0
    16   30           0 DEFR4K READSB       11 - READ SUB (APPEND PRGM)
    16   31           0
    17   32           0 DEFR4K RENAME       12 - RENAME A CASSETTE FILE
    17   33           0
    18   34           0 DEFR4K SEC          13 - SECURE FILE PROTECTION
    18   35           0
    19   36           0 DEFR4K SEEKR        14 - SEEK TO A REG. IN FILE
    19   37           0
    20   40           0 DEFR4K UNSEC        15 - REMOVE FILE PROTECTION
    20   41           0
    21   42           0 DEFR4K VERIFY       16 - VERIFY A READABLE FILE
    21   43           0
    22   44           0 DEFR4K WRTA         17 - WRITE ALL INFO TO FILE
    22   45           0
    23   46           0 DEFR4K WRTK         18 - WRITE "KEY" ASSIGNMENT
    23   47           0
    24   50           0 DEFR4K WRTP         19 - WRITE PROGRAM NORMALLY
    24   51           0
    25   52           0 DEFR4K WRTPV        20 - WRITE PROGRAM: PRIVATE
    25   53           0
    26   54           0 DEFR4K WRTR         21 - WRITE A REGISTER BLOCK
    26   55           0
    27   56           0 DEFR4K WRTRX        22 - WRITE REGS USING X-REG
    27   57           0
    28   60           0 DEFR4K WRTS         23 - WRITE STATUS TO MEDIUM
    28   61           0
    29   62           0 DEFR4K ZERO         24 - FILL DATA FILE W/ZEROS
    29   63           0
    30   64           0 DEFR4K CSTNOP       25 - NULL CASSETTE FUNCTION
    30   65           0
```

```
   31   66           0 DEFR4K PILPRM        26 - SUBGROUP ID: -CTL FCNS
   31   67           0
   32   70           0 DEFR4K AUTOIO        27 - AUTO MODE: CLR FLAG 32
   32   71           0
   33   72           0 DEFR4K FINDID        28 - FIND DEVICE LAD VIA ID
   33   73           0
   34   74           0 DEFR4K INA           29 - INPUT STRING ALPHA REG
   34   75           0
   35   76           0 DEFR4K INDX          30 - INPUT DECIMAL TO X-REG
   35   77           0
   36  100           0 DEFR4K INSTAT        31 - INPUT DEV. STATUS INFO
   36  101           0
   37  102           0 DEFR4K LISTNR        32 - ADDR DEV AS A LISTENER
   37  103           0
   38  104           0 DEFR4K LOCAL         33 - PRIM DEV TO LOCAL MODE
   38  105           0
   39  106           0 DEFR4K MANIO         34 - MANUAL MODE: SET FL 32
   39  107           0
   40  110           0 DEFR4K OUTA          35 - OUTPUT ALPHA  REGISTER
   40  111           0
   41  112           0 DEFR4K PWRDN         36 - SEND: "GROUP PWR DOWN"
   41  113           0
   42  114           0 DEFR4K PWRUP         37 - SEND: "GROUP POWER UP"
   42  115           0
   43  116           0 DEFR4K REMOTE        38 - PRIMARY TO REMOTE MODE
   43  117           0
   44  120           0 DEFR4K SELECT        39 - SELECT THE DEVICE BY X
   44  121           0
   45  122           0 DEFR4K STOPIO        40 - INTERFACE CLEAR & HALT
   45  123           0
   46  124           0 DEFR4K TRIGER        41 - EXEC GROUP TRIGGER CMD
   46  125           0
   47  126           0 CON    0                  NULL
   48  127           0 CON    0                  NULL
   49  130         223 CON    @223           S
   50  131          16 CON    @16            N
   51  132           6 CON    @06            F
   52  133          40 CON    @40                BLANK
   53  134          14 CON    @14            L
   54  135          24 CON    @24            T
   55  136           3 CON    @03            C
   56  137          55 CON    @55            -
   57      PILPRM      ENTRY  PILPRM
   58
   59  140         255 CON    @255           -
   60  141          55 CON    @55            -
   61                 ENTRY  CSTNOP
   62  142 CSTNOP 1740 RTN                   END OF DUMMY CASSETTE FN
*************************************************************************
* PILEN - ROUTINE TO ENABLE PIL CHIP                                    *
*         1. WRITE TO R3R/W TO TURN ON OSCILLATOR                       *
*         2. SET MASTER CLEAR                                           *
*         3. CLEAR MASTER CLEAR                                         *
*         4. ADDRESS SELF AS A SC, CA, LA                               *
*         5. SEND OUT "ASP 1"                                           *
* USES:   A[X], C, +0 SUBROUTINE LEVELS                                 *
* OUTPUT: A[X] = NUMBER OF DEVICES IN THE LOOP                          *
*************************************************************************
   73                 ENTRY  PILEN
   74                 ENTRY  ASP
```

```
*
    76  143 PILEN   116 C=0                 CLEAR ACCUMULATOR
    77  144        1110 S9=    1            SET ERROR FLAG
    78  145        1160 DADD=C              ENABLE CHIP 0
    79  146        1670 C=REGN 14           LOAD FLAGS REGISTER
    80  147         574 RCR    6            C[13]=FLAGS 32-35
    81  150         776 C=C+C  S            SHIFT FLAG 32 TO CARRY
    82  151         776 C=C+C  S            USER FLAG 33 SET ?
    83  152        1540 RTN C               YES, DON'T DO ANYTHING
    84  153         344 C=HPIL 3            TURN ON CLOCK - P.P. REG
    84  154         372                     (INSERTED BY ASSEMBLER)
    84  155         303                     (INSERTED BY ASSEMBLER)
    85  156        1730 CST EX              PARA POLL BITS TO STATUS
    86  157        1204 S7=    0            CLEAR OSCILLATOR DISABLE
    87  160        1730 CST EX              PARA POLL BITS TO C[X]
    88  161        1300 HPIL=C 3            WRITE PARALLEL POLL REG
    89  162          44 HPL=CH 0            WRITE STATUS REGISTER
    90  163           5 CH=    @001         SET MASTER CLEAR
    91  164          44 HPL=CH 0            WRITE STATUS REGISTER
    92  165        1611 CH=    @342         SC=CA=TA=1
    93  166        1104 S9=    0            CLEAR ERROR FLAG
* SEND OUT AN IDY TO CHECK THE LOOP INTEGRITY.
* TIME OUT ONLY SET FOR 1.21 SECONDS.
    96  167         144 HPL=CH 1            WRITE CONTROL INT REGISTER
    97  170        1405 CH=    @301         IDENTIFY CLASS (BITS 7 & 6)
    98  171        1200 HPIL=C 2            WRITE DATA BITS REGISTER
    99  172         460 LDI                 LOAD LOW 12 BITS OF C WITH
   100  173        1750 CON    1000         C[X] = 1000
   101  174         746 C=C+C  X            C[X] = 2000
   102  175 IDYTST 1146 C=C-1  X            TIME OUT YET ?
   103  176         247 GOC    AAU25  ( 222) YES, SOMETHING WENT WRONG
   104  177         354 ORAV?               IDY COMES BACK YET ?
   105  200        1753 GONC   IDYTST ( 175) NOT YET, KEEP CHECKING
   106  201         244 C=HPIL 2            READ DATA BITS REGISTER
   106  202         272                     (INSERTED BY ASSEMBLER)
   106  203         203                     (INSERTED BY ASSEMBLER)
   107  204           0 NOP                 DO AUTO ADDRESSING ALWAYS
   108  205 ASP     144 HPL=CH 1            WRITE CONTROL INT REGISTER
   109  206        1005 CH=    @201         COMMAND CLASS (BIT 7)
   110  207         244 HPL=CH 2            WRITE DATA BITS REGISTER
   111  210        1151 CH=    @232         AAU - AUTO ADDR UNCONFIGURE
   112  211         106 C=0    X            SET COUNTER TO 0
   113  212 AAU10   354 ORAV?               OUTPUT REGISTER AVAILABLE ?
   114  213          57 GOC    AAU20  ( 220) YES, OUTPUT REG AVAILABLE
   115  214           0 NOP                 NO, REG NOT AVAILABLE YET
   116  215        1046 C=C+1  X            TIME OUT YET ?
   117  216          47 GOC    AAU25  ( 222) YES, ERROR IF C[X] > FFF
   118  217        1733 GOTO   AAU10  ( 212) NO, KEEP TRYING OUTPUT REG
   119  220 AAU20  1154 FRNS?               FRAME RETURN SAME AS SENT ?
   120  221          33 GONC   ASP00  ( 224) YES, SAME FRAME RETURNED
   121  222 AAU25     6 A=0    X            NO, SOMETHING WENT WRONG
   122  223         633 GOTO   SCMDER ( 306) INDICATE AN ERROR
   123  224 ASP00   144 HPL=CH 1            WRITE CONTROL INT REGISTER
   124  225        1205 CH=    @241         READY CLASS (BITS 7 & 5)
   125  226         244 HPL=CH 2            WRITE DATA BITS REGISTER
   126  227        1005 CH=    @201         ASP 1 (ASSIGN ADDRS FROM 1)
   127  230         106 C=0    X            INITIALIZE TIMER
   128                 ENTRY  ASP10
   129  231 ASP10   354 ORAV?               FRAME COMES BACK YET?
   130  232          57 GOC    ASP30  ( 237) YES, FRAME HAS RETURNED
```

```
   131  233           0 NOP                    NO, REGISTER STILL BUSY
   132  234        1046 C=C+1  X               TIME OUT YET ?
   133  235        1657 GOC    AAU25  ( 222) YES, ERROR IF C[X] > FFF
   134  236        1733 GOTO   ASP10  ( 231) NOT YET, KEEP TRYING
   135  237 ASP30  1154 FRNS?                  FRAME RETURN SAME AS SENT ?
   136  240          63 GONC   ASP50  ( 246) YES, NOBODY OUT THERE
   137  241 ASP40   244 C=HPIL 2               READ THE RETURN FRAME
   137  242         272                       (INSERTED BY ASSEMBLER)
   137  243         203                       (INSERTED BY ASSEMBLER)
   138  244        1146 C=C-1  X               C[X]= # OF DEV ON LOOP
   139  245          23 GONC   ASP60  ( 247) SAVE NUMBER OF DEVICES
   140  246 ASP50   106 C=0    X               NO DEVICES ON LOOP
   141  247 ASP60  1730 CST EX                 MOVE C[1:0] TO STATUS
   142  250        1204 S7=    0               SWITCH OFF BIT 7
   143  251        1730 CST EX                 MOVE STATUS TO C[1:0]
   144  252         406 A=C    X               A[X] = # OF DEVICES
   145  253        1740 RTN                    END OF ENABLE PIL CHIP
*********************************************************************
* SCMD - ROUTINE TO SEND OUT A COMMAND                               *
*                                                                    *
* USES C ONLY                                                        *
* SUBENTRIES : UNL    - SEND UNLISTEN COMMAND                        *
*              UNT    - SEND UNTALK COMMAND                          *
*              TAD    - SEND TALKER ADDRESS (TALKER ADDRESS IN C[X]) *
*              LAD    - SEND LISTENER ADDR (LISTENER ADDRESS IN C[X])*
*              LADA   - SAME AS LAD EXCEPT LISTENER IN A[X] (REMOVED?)*
*              SNDATA - SEND SEC. CMD - "SEND DATA"                  *
*              TALKER - ADDRESS DEVICE AS A TALKER (TAD IN R5)       *
*              LISTEN - ADDRESS DEVICE AS LISTENER (LAD IN R6)       *
*              SRCHFL - MAKE IT A LISTENER & SEND "SEARCH" (REMOVED?)*
*              DTFLOW - SEND SEC. CMD - "DATA FOLLOW"                *
*                                                                    *
* TIME OUT = 4.4 SECONDS FOR ALL FOLLOWING COMMAND FRAMES            *
*                                                                    *
*********************************************************************
   164                   ENTRY  TALKER
*
   166                   ENTRY  SCMD
   167                   ENTRY  SCMD15
   168                   ENTRY  SCMD20
   169                   ENTRY  SCMD30
   170                   ENTRY  SCMDER
   171                   ENTRY  SFRMC
   172                   ENTRY  TAD
   173                   ENTRY  UNL
   174                   ENTRY  UNT
*
   176  254 UNT     460 LDI                    SEND UNTALK COMMAND
   177  255         137 CON    @137            @137 = UNTALK
   178  256         143 GOTO   SCMD   ( 272) SEND IT
   179  257 UNL     460 LDI                    SEND UNLISTEN COMMAND
   180  260          77 CON    @077            @077 = UNLISTEN
   181  261         113 GOTO   SCMD   ( 272) SEND IT
   182  262 TALKER  544 C=HPIL 5               GET DEVICE ADDRESS
   182  263         572                       (INSERTED BY ASSEMBLER)
   182  264         503                       (INSERTED BY ASSEMBLER)
   183  265 TAD      44 HPL=CH 0               WRITE STATUS REGISTER
   184  266        1501 CH=    @320            SET SC, CA, LA = 1
   185  267        1730 CST EX                DEV ADDR IN C[X] TO STATUS
   186  270         510 S6=    1               TURN ADDR N INTO TAD.N
```

```
   187  271         1730 CST EX                C[X] NOW HAS TALKER ADDR
   188  272 SCMD     144 HPL=CH 1              WRITE CONTROL INT REGISTER
   189  273         1005 CH=    @201           COMMAND CLASS (BIT 7)
   190  274 SFRMC   1200 HPIL=C 2             WRITE BYTE C[1:0] TO R2W
   191  275 SCMD15   106 C=0     X             INITIALIZE TIMER *CAUTION*
   192  276 SCMD20   354 ORAV?                 FRAME COMES BACK YET ?
   193  277          167 GOC     SCMD30 ( 315) YES, CHECK FRAME INTEGRITY
   194  300            0 NOP                   WAIT
   195  301            0 NOP                   WAIT
   196  302            0 NOP                   WAIT
   197  303            0 NOP                   WAIT
   198  304         1046 C=C+1   X             TIME OUT YET ?
   199  305         1713 GONC    SCMD20 ( 276) NOT YET, KEEP CHECKING
   200  306 SCMDER 1110 S9=     1              SET ERROR FLAG
   201  307          244 C=HPIL 2              RESET FRNS
   201  310          272                       (INSERTED BY ASSEMBLER)
   201  311          203                       (INSERTED BY ASSEMBLER)
   202  312 SCMD26   144 HPL=CH 1             WRITE CONTROL INT REGISTER
   203  313            1 CH=    @000           TURN OFF FLAG ENABLE
   204  314         1740 RTN                   RETURN FROM SCMD ROUTINE
   205  315 SCMD30 1154 FRNS?                  FRAME RETURN NOT AS SENT ?
   206  316         1743 GONC    SCMD26 ( 312) NO, NO TRANSMIT ERROR
   207  317         1673 GOTO    SCMDER ( 306) YES, INDICATE AN ERROR
   208                   ENTRY   SNDATA
   209                   ENTRY   DTFLOW
   210                   ENTRY   LISTEN
   211                   ENTRY   WRTDAT
   212                   ENTRY   SEKSUB
   213                   ENTRY   SRWRT
   214                   ENTRY   DDT0
*
   216  320 DDT0     460 LDI                   DEVICE DEPENDENT TALKER 0
   217  321          300 CON     @300          DDT 0 = READ BUFFER 0
   218  322         1503 GOTO    SCMD   ( 272) SEND IT
   219  323 SRWRT    460 LDI                   DEVICE DEPENDENT LISTENER 2
   220  324          242 CON     @242          DDL 2 = WRITE
   221  325         1453 GOTO    SCMD   ( 272) SEND IT
   222  326 SEKSUB     1 GOSUB   SEEKN         SEEK TO RECORD, REC # IN N
   222  327            0                       *ILCAS&CTL:  CS3, @1562
   223  330 WRTDAT     1 GOSUB   SRWRT         SEND DDL 2 COMMAND (WRITE)
   223  331            0                       *ILCAS&CTL:  CS0, @0323
   224  332 DTFLOW   460 LDI                   DEVICE DEPENDENT LISTENER 0
   225  333          240 CON     @240          DDL0 = WRITE BUFFER 0
   226  334         1363 GOTO    SCMD   ( 272) SEND IT
   227  335 LISTEN    44 HPL=CH 0              WRITE STATUS REGISTER
   228  336         1601 CH=    @340           SELF AS SC, CA, TA
   229  337          544 C=HPIL 5              GET DEVICE NUMBER
   229  340          572                       (INSERTED BY ASSEMBLER)
   229  341          503                       (INSERTED BY ASSEMBLER)
   230                   ENTRY   LAD
   231  342 LAD     1730 CST EX                DEV ADDR IN C[X] TO STATUS
   232  343          210 S5=     1             MAKE IT A LISTENER ADDRESS
   233  344         1730 CST EX                C[X] NOW HAS LISTENER ADDR
   234  345         1253 GOTO    SCMD   ( 272) SEND IT
   235  346 SNDATA   460 LDI                   DEVICE DEPENDENT TALKER 0
   236  347          300 CON     @300          DDT0 = READ BUFFER 0
   237  350            1 GOSUB   SCMD          SEND COMMAND TO DEVICE
   237  351            0                       *ILCAS&CTL:  CS0, @0272
* FALL INTO ROUTINE TO SEND NAT
*********************************************************************
```

```
* NATNRD - SEND THE READY FRAME "NAT"                                  *
*          AFTER SENDING OUT THE NAT, WAIT FOR THE ORAV TO GET SET.    *
*          MEANWHILE, CHECK THE KEYBOARD. ANY KEY DOWN WILL CAUSE      *
*          PROGRAM TO ABORT THE WAIT LOOP AND RETURN TO MAINFRAME.     *
*          AS SOON AS ORAV SET, RETURN TO CALLING PROGRAM WITHOUT      *
*          READING THE INCOMING FRAME.                                 *
************************************************************************
   247                  ENTRY  NATNRD
*
   249  352 NATNRD  144 HPL=CH 1             WRITE CONTROL INT REGISTER
   250  353        1205 CH=    @241          READY CLASS (BITS 7 & 5)
   251  354         244 HPL=CH 2             WRITE DATA BITS REGISTER
   252  355         601 CH=    @140          SEND NAT (SDA = SEND DATA)
   253  356         106 C=0    X             INITIALIZE TIMER
   254  357 NATN10  454 FRAV?                ANY FRAME COMES IN YET?
   255  360        1540 RTN C                YES, ROUTINE IS FINISHED
   256  361        1046 C=C+1  X             TIME OUT YET ?
   257  362         547 GOC    RDFMER ( 436) YES, INDICATE AN ERROR
   258  363        1743 GOTO   NATN10 ( 357) NO, NOT YET, KEEP TRYING
*
************************************************************************
* NRD  - SEND NOT READY FRAME                                          *
*        1. READ THE LAST FRAME AND SAVE IT IN R6, BUT DON'T ECHO IT.  *
*        2. THEN SEND "NRD".                                           *
*        3. AFTER NRD RETURNS, RETRANSMIT THE LAST DATA FRAME.         *
*        4. THEN READ ETO/ETE                                          *
* NRDC - SAME AS NRD EXCEPT R2 ALREADY IN C[X]                         *
* ASSUMES: NOTHING                                                     *
* OUTPUT : C[X] = LAST FRAME                                           *
* USES   : A[X], C, +1 SUBROUTINE LEVEL                                *
************************************************************************
   271                  ENTRY  NRD
   272                  ENTRY  NRDC
*
   274  364 NRD       1 GOSUB  RDDFRM         READ IN A DATA FRAME
   274  365           0                       *ILCAS&CTL:  CS0, @0420
   275  366 NRDC    406 A=C    X              SAVE R2 IN A[X]
   276  367         144 HPL=CH 1              WRITE READY CONTROL BITS
   277  370        1205 CH=    @241           READY CLASS (BITS 7 & 5)
   278  371         460 LDI                   SEND NRD
   279  372         102 CON    @102           NOT READY FOR DATA COMMAND
   280  373           1 GOSUB  SFRMC          SEND FRAME C[1:0] TO R2W
   280  374           0                       *ILCAS&CTL:  CS0, @0274
   281  375         244 C=HPIL 2              GET NRD OUT OF INPUT BUFFER
   281  376         272                       (INSERTED BY ASSEMBLER)
   281  377         203                       (INSERTED BY ASSEMBLER)
   282  400         144 HPL=CH 1              WRITE CONTROL INT REGISTER
   283  401           5 CH=    @001           ENABLE FI LINE (BIT 0)
   284  402         246 C=A    X              LAST DATA FRAME FROM A[X]
   284  403         406                       (INSERTED BY ASSEMBLER)
   285  404        1200 HPIL=C 2              RETRANSMIT LAST DATA FRAME
   286  405           1 GOSUB  RDDFRM         READ ETO/ETE
   286  406           0                       *ILCAS&CTL:  CS0, @0420
   287  407        1104 S9=    0              CLEAR ERROR FLAG
   288  410         246 AC EX  X              C[X] = LAST DATA FRAME
   289  411        1074 RCR    2              A[X] = ETO/ETE
   290  412         460 LDI                   END TRANSMISSION OK
   291  413         100 CON    @100           @100 = ETO COMMAND
   292  414        1546 ? A#C  X              IS IT AN ETO ?
   293  415         217 GOC    RDFMER ( 436) NO, IT MUST BE AN ETE
```

```
   294  416        1574 RCR    12            C[X] = LAST DATA FRAME
   295  417        1740 RTN                  END OF SEND NOT READY FRAME
*************************************************************************
* RDDFRM - READ A DATA FRAME                                            *
*                                                                       *
* INPUT:   ASSUMES THE DEVICE IS A LISTENER                             *
*          IF S9 = 1 WILL RETURN WITHOUT TRYING                         *
* OUTPUT:  C[1:0] = DATA BYTE                                           *
*          IF TIME OUT WILL SET C=0 & S9=1                              *
*          IF THE FRAME IS NOT A DATA FRAME WILL SET S9=1               *
* NOTE:    THE DATA FRAME WILL NOT BE ECHOED BY THIS ROUTINE            *
* USES:    C, +0 SUBROUTINE LEVELS                                      *
*************************************************************************
   307                  ENTRY  RDDFRM
   308                  ENTRY  RDFMER
   309                  ENTRY  C=R2
   310                  ENTRY  UNLRSF
*
   312  420 RDDFRM  116 C=0                   INITIALIZE TIME OUT COUNTER
   313  421 RDFM10  454 FRAV?                 HAS ANY FRAME COME IN YET ?
   314  422          67 GOC    RDFM20 ( 430) YES, HANDLE THE FRAME
   315  423        1046 C=C+1  X               TIME OUT YET ?
   316  424        1753 GONC   RDFM10 ( 421) NO, KEEP LOOKING FOR FRAME
   317  425        1076 C=C+1  S               INC SIGN DIGIT - TIME OUT ?
   318  426        1733 GONC   RDFM10 ( 421) NO, KEEP LOOKING FOR FRAME
   319  427          73 GOTO   RDFMER ( 436) YES, TIME OUT FINALLY
   320  430 RDFM20  144 C=HPIL 1               READ CONTROL BITS
   320  431         172                       (INSERTED BY ASSEMBLER)
   320  432         103                       (INSERTED BY ASSEMBLER)
   321  433        1074 RCR    2               MOVE C[1] TO C[13] (SIGN)
   322  434         776 C=C+C  S               IS THIS A DATA FRAME?
   323  435          23 GONC   C=R2   ( 437) YES, PROCESS DATA FRAME
   324  436 RDFMER 1110 S9=    1               SET ERROR FLAG
   325  437 C=R2    244 C=HPIL 2               READ DATA BITS
   325  440         272                       (INSERTED BY ASSEMBLER)
   325  441         203                       (INSERTED BY ASSEMBLER)
   326  442        1740 RTN                   RETURN AFTER READING DATA
   327  443 UNLRSF    1 GOSUB  UNL            SEND UNLISTEN COMMAND
   327  444          0                       *ILCAS&CTL:  CS0, @0257
   328  445        1723 GOTO   C=R2   ( 437) PROCESS DATA FRAME
*************************************************************************
* SDATA  - SEND OUT A DATA FRAME AND SET TIME OUT TO 40 SECONDS         *
* INPUT:   C[1:0] = DATA BYTE                                           *
* OUTPUT:  S9     = 1 IF TIME OUT OR "FRNS" SET                         *
* USES:    C, +0 SUBROUTINE LEVELS                                      *
*************************************************************************
   335                  ENTRY  SDATA0
   336                  ENTRY  SDATA
*
   338  446 SDATA0  144 HPL=CH 1               WRITE CONTROL INT REGISTER
   339  447          5 CH=    @001             ENABLE FI LINE (BIT 0)
   340  450 SDATA  1200 HPIL=C 2               WRITE DATA REGISTER
   341  451         116 C=0                    INIT TIME OUT COUNTER
   342  452 SDAT10  354 ORAV?                  OUTPUT REG AVAILABLE ?
   343  453          67 GOC    SDAT20 ( 461) YES, PROCESS THE DATA
   344  454        1046 C=C+1  X               TIME OUT ?
   345  455        1753 GONC   SDAT10 ( 452) NO, LOOK FOR OUTPUT REG
   346  456        1076 C=C+1  S               INC SIGN DIGIT - TIME OUT ?
   347  457        1733 GONC   SDAT10 ( 452) NO, LOOK FOR OUTPUT REG
   348  460        1563 GOTO   RDFMER ( 436) YES, TIME OUT ERROR
```

```
   349  461 SDAT20 1154 FRNS?                FRAME RETURN SAME AS SENT ?
   350  462        1640 RTN NC               YES, RETURN WITHOUT ERROR
   351  463        1533 GOTO   RDFMER ( 436) NO, BAD FRAME ERROR
*******************************************************************************
* RDTYPE - READ A DEVICE TYPE (SAC = SAI = SEND ACCESSORY CODE OR ID)  *
*          ASSUMES THE DEVICE ALREADY ADDRESSED AS A TALKER, SEND RDY  *
*          FRAME "SAC" AND READ ONE DATA BYTE FROM IT.  IF THE DEVICE  *
*          SENDS MORE THAN ONE BYTE, ONLY THE FIRST BYTE WILL BE KEPT. *
*                                                                      *
* USES:    A[X], C, S0, NO PT, +0 SUBROUTINE LEVELS                    *
* OUTPUT:  A[2:1] = STATUS BYTE IF THE DEVICE RESPONDS TO SAC          *
*          A[2:1] = 0 IF DEVICE DOES NOT RESPOND TO RDY FRAME - "SAC"  *
*                                                                      *
* DEVICE TYPE HAS BEEN DEFINED AS FOLLOWS:                             *
* FILBERT   - HEX 10 (HP-82161A CASSETTE DRIVE)                        *
* SPECIAL.K - HEX 20 (HP-82162A THERMAL PRINTER)                       *
* WALLABY   - HEX 30 (HP-82163A VIDEO INTERFACE)                       *
* PIL.BOX   - HEX 40 (HP-82166A HP-IL CONVERTER)                       *
* PLOTTER   - HEX 50 (HP-7470A GRAPHICS PLOTTER)                       *
*******************************************************************************
   369                  ENTRY  RDTYPE
   370                  ENTRY  RDTYPC
   371                  ENTRY  RTPHRT
*
   373  464 RDTYPE  460 LDI                  SEND SAC COMMAND
   374  465         143 CON    @143          @143 = SAC/SAI COMMAND
   375  466 RDTYPC  144 HPL=CH 1             WRITE CONTROL INT REGISTER
   376  467        1205 CH=    @241          READY CLASS (BITS 7 & 5)
   377  470        1200 HPIL=C 2             WRITE SAC/SAI COMMAND
   378  471           6 A=0    X             CLEAR REGISTER "A"
   379  472        1610 S0=    1             SET TO INDICATE FIRST BYTE
   380  473 RTYP10  106 C=0    X             INITIALIZE TIMER
   381  474 RTYP20  354 ORAV?                DID THE FRAME RETURN YET ?
   382  475          57 GOC    RTYP30 ( 502) YES, CHECK INCOMING FRAME
   383  476           0 NOP                  NO, REGISTER STILL BUSY
   384  477        1046 C=C+1  X             TIME OUT ?
   385  500        1367 GOC    RDFMER ( 436) YES, TIME OUT ERROR
   386  501        1733 GOTO   RTYP20 ( 474) NO, NOT YET, KEEP CHECKING
   387  502 RTYP30  144 C=HPIL 1             CHECK INCOMING DATA FRAME
   387  503         172                      (INSERTED BY ASSEMBLER)
   387  504         103                      (INSERTED BY ASSEMBLER)
   388  505        1374 RCR    13            ROTATE LEFT 1 DIGIT
   389  506         746 C=C+C  X             CHECK TOP DIGIT OF C[X]
   390  507        1307 GOC    C=R2   ( 437) NOT DATA FRAME, READ R2
   391  510           1 GOLONG RTPACH        READ THE RETURN DATA BYTE
   391  511           2                      *ILCAS&CTL:  CS0, @0671
   392  512 RTPHRT 1614 ?S0=1                FIRST BYTE ?
   393  513        1603 GONC   RTYP10 ( 473) NO, IGNORE IT
   394  514         406 A=C    X             SAVE FIRST BYTE TO A[X]
   395  515        1746 A SL   X             SHIFT A[X] LEFT 1 DIGIT
   396  516        1604 S0=    0             FIRST BYTE ALREADY FOUND
   397  517        1543 GOTO   RTYP10 ( 473) READ NEXT FRAME
*******************************************************************************
* GETDEV - GET SELECTED OR DEFAULT DEVICE NUMBER                       *
*                                                                      *
* LOGIC:   USE SELECTED DEVICE # OR DEFAULT TO 1 IF NO DEV SELECTED.   *
*                                                                      *
* INPUT:   A[X]  = NUMBER OF DEVICES IN LOOP                           *
*                                                                      *
* OUTPUT:  C[X]  = START SEARCHING DEVICE NUMBER                       *
```

```
*          R5R/W = START SEARCHING DEVICE NUMBER                        *
*          R6R/W = R5R/W IF STARTING DEV # <= # OF DEVICES IN LOOP      *
*                = R5R/W = 1 IF STARTING DEV # > # DEVICES IN LOOP      *
*                                                                       *
* USES:    C, S[7:0], +0 SUBROUTINE LEVELS                              *
*************************************************************************
   412                  ENTRY  GETDEV
*
   414  520 GETDEV  106 C=0    X             CLEAR C[2:0]
   415  521        1160 DADD=C                ENABLE CHIP 0
   416  522        1670 C=REGN 14             LOAD FLAGS REGISTER
   417  523         274 RCR    5              C[1:0] = FLAGS 28-35
   418  524        1530 ST=C                  S3 = FLAG 32 (NOT TESTED!)
   419  525         444 C=HPIL 4              READ LOOP ADDRESS REGISTER
   419  526         472                       (INSERTED BY ASSEMBLER)
   419  527         403                       (INSERTED BY ASSEMBLER)
   420  530        1406 ? A<C  X              START # > # OF DEVICES ?
   421  531          33 GONC   GTD#20 ( 534) NO, END # = START #
   422  532         106 C=0    X              YES, END # = START # = 1
   423  533        1046 C=C+1  X              SET START/END VALUE TO 1
   424  534 GTD#20 1500 HPIL=C 5              WRITE DEVICE # REGISTER 5
   425  535        1600 HPIL=C 6              WRITE DEVICE # REGISTER 6
   426  536        1740 RTN                   END OF GET DEVICE ROUTINE
   427                  FILLTO @540
        537        0000 NOP
        540        0000 NOP
*************************************************************************
* NXTDEV - ROUTINE TO GET NEXT DEVICE NUMBER                            *
*                                                                       *
* LOGIC :  1. IF IN UNFRIENDLY MODE, RETURN TO P+1 IMMEDIATELY,         *
*             DEVICE NOT FOUND.                                         *
*          2. IF CURRENT SEARCHING DEVICE # < # OF DEVICES IN LOOP,     *
*             NEXT DEVICE NUMBER = CURRENT DEVICE + 1                   *
*          3. IF CURRENT SEARCHING DEVICE # >= # OF DEVICES IN LOOP,    *
*             NEXT DEVICE NUMBER = 1                                    *
*          4. IF NEXT DEVICE # NOT EQUAL TO START SEARCHING DEVICE #,   *
*             DEVICE NOT FOUND, RETURN TO P+1                           *
*          5. IF NEXT DEVICE NOT EQUAL TO STARTING DEVICE,              *
*             RETURN TO P+2 WITH NEXT DEVICE NUMBER IN C[X]             *
*                                                                       *
* INPUT:   A[X]  = NUMBER OF DEVICES IN THE LOOP                        *
*          R5R/W = DEVICE NUMBER CURRENTLY BEING POLLED                 *
*          R6R/W = START SEARCHING DEVICE NUMBER                        *
* OUTPUT:  C[X]  = NEXT SEARCHING DEVICE NUMBER                         *
* USES:    A[X], C, S[7:0], +0 SUBROUTINE LEVELS                        *
*************************************************************************
   448                  ENTRY  NXTDEV
   449                  ENTRY  NXTDEI
*
   451  541 NXTDEV 1670 C=REGN 14             LOAD FLAGS REGISTER
   452  542         574 RCR    6              C[S]=FLAGS 32-35
   453  543         776 C=C+C  S              IN UNFRIENDLY MODE ?
   454  544        1540 RTN C                 YES, DON'T GOTO NEXT DEV
   455  545 NXTDEI  544 C=HPIL 5              GET CURRENT DEVICE NUMBER
   455  546         572                       (INSERTED BY ASSEMBLER)
   455  547         503                       (INSERTED BY ASSEMBLER)
   456  550 NXTD25 1046 C=C+1  X              POINT TO NEXT DEV #
   457  551        1406 ? A<C  X              WRAP AROUND THE LOOP YET ?
   458  552          33 GONC   NXTD30 ( 555) NOT YET
   459  553         106 C=0    X              YES, START OVER FROM DEV #1
```

```
   460  554         1046 C=C+1  X             SET CURRENT DEVICE # TO 1
   461  555 NXTD30   406 A=C    X             A[X] = CURRENT DEVICE #
   462  556          644 C=HPIL 6             GET STARTING DEVICE #
   462  557          672                      (INSERTED BY ASSEMBLER)
   462  560          603                      (INSERTED BY ASSEMBLER)
   463  561          246 AC EX  X             SWAP CURR/START DEVICE #
   464  562         1500 HPIL=C 5             WRITE CURR DEV # TO R5R/W
   465  563         1546 ? A#C  X             SEARCH ENTIRE LOOP YET ?
   466  564         1640 RTN NC               YES, DEVICE NOT FOUND
   467  565          713 GOTO   RTNP+2 ( 656) RETURN TO CALL ADDR + 2
*******************************************************************************
* FNDPTR - FIND A PRINTER IN A PIL LOOP                                       *
* FNDCAS - FIND A CASSETTE IN A LOOP                                          *
*                                                                             *
* ASSUMES: CHIP 0 ENABLED                                                     *
* USES:    A, C, S[7:0], NO PT, +1 SUB LEVEL                                  *
* OUTPUT:                                                                     *
*          IF PRINTER OR CASSETTE FOUND:                                      *
*          1. RETURN TO P+2                                                   *
*          2. SAVE DEVICE ADDRESS IN R5R/W                                    *
*          3. S9 = 0                                                          *
*          IF PRINTER OR CASSETTE NOT FOUND:                                  *
*          1. RETURN TO P+1                                                   *
*          2. S9 = 1                                                          *
*******************************************************************************
   483                   ENTRY  FNDPTR
   484                   ENTRY  FNDCAS
   485                   ENTRY  RTNP+2
   486                   ENTRY  FDEV20
*
   488  566 FNDCAS     1 GOSUB  PILEN           ENABLE PIL CHIP
   488  567            0                        *ILCAS&CTL:  CS0, @0143
   489  570           36 A=0    S               CLEAR A[S] DIGIT
   490  571          576 A=A+1  S               SET A[S] TO 1
   491                   LEGAL                  (CLEAR THE CARRY FLAG)
   492  572            1 GOSUB  PLERCK          CHECK IF LOOP IS INTACT
   492  573            0                        *ILCAS&CTL:  CS1, @1747
   493  574           53 GOTO   FNDDEV ( 601)   FINISH FINDING CASSETTE
   494  575 FNDPTR   36 A=0    S               CLEAR A[S] SIGN DIGIT
   495  576         1140 SETHEX                ENTER HEXADECIMAL MODE
   496  577            1 GOSUB  PILEN           ENABLE PIL CHIP
   496  600            0                        *ILCAS&CTL:  CS0, @0143
   497  601 FNDDEV 1114 ?S9=1                  FRAME GO THRU THE LOOP ?
   498  602         1540 RTN C                 NO, STOP LOOKING FOR DEV
   499  603          674 RCR    11             C[M] = NUMBER OF DEVICES
   500  604          432 A=C    M              A[M] = NUMBER OF DEVICES
   501  605            1 GOSUB  GETDEV          GET START SEARCHING DEV#
   501  606            0                        *ILCAS&CTL:  CS0, @0520
   502  607 FDEV10     1 GOSUB  TAD             ADDRESS DEVICE AS A TALKER
   502  610            0                        *ILCAS&CTL:  CS0, @0265
   503  611            1 GOSUB  RDTYPE          READ THE DEVICE TYPE
   503  612            0                        *ILCAS&CTL:  CS0, @0464
   504  613          460 LDI                    LOAD LOW 12 BITS OF C WITH
   505  614          400 CON    @400            C[X] = @400 = HEX 100
   506  615         1536 ? A#0  S               LOOKING FOR PRINTER ?
   507  616          133 GONC   FRTR35 ( 631)   YES, SET C[X] = HEX 200
   508  617         1546 ? A#C  X               IS IT FILBERT TYPE MASS ST?
   509  620          323 GONC   FDEV40 ( 652)   YES, SIMILAR TO FILBERT
   510  621 FDEV20   256 C=A                    GET # OF DEVICES IN LOOP
   510  622          416                        (INSERTED BY ASSEMBLER)
```

```
   511  623          74 RCR    3             C[X] = # OF DEVS IN LOOP
   512  624         406 A=C    X             A[X] = # OF DEVS IN LOOP
   513  625           1 GOSUB  NXTDEV        GET NEXT DEV # & CHK DONE
   513  626           0                       *ILCAS&CTL:  CS0, @0541
   514  627         323 GOTO   FDEV50 ( 661) P+1 - DEVICE NOT FOUND
   515  630        1573 GOTO   FDEV10 ( 607) P+2 - EXAMINE NEXT DEVICE
   516  631 FRTR35  746 C=C+C  X             C[X] = HEX 200
   517  632        1546 ? A#C  X             IS IT THE SPECIAL-K ?
   518  633         113 GONC   FDEV38 ( 644) YES, HP-82162A PRINTER
   519  634        1566 ? A#C  XS            OTHER KIND OF PRINTER ?
   520  635          43 GONC   FDEV36 ( 641) YES, OTHER PRINTER
   521  636        1066 C=C+1  XS            C[XS] = 3
   522  637        1566 ? A#C  XS            SOME DISPLAY ?
   523  640        1617 GOC    FDEV20 ( 621) NO, GO BACK, TRY AGAIN
   524  641 FDEV36  106 C=0    X             YES, DON'T ASK ITS STATUS
   525  642        1146 C=C-1  X             ALL 1S IN R6 SAYS T.V.
   526  643        1600 HPIL=C 6             WRITE FF TO R6R/W
   527  644 FDEV38    1 GOSUB  LDSST0        SEE IF PRT EXISTS FLAG SET
   527  645           0                       *MAINFRAME:  CN1, @1627
   528  646        1614 ?S0=1                CHECK USER FLAG 55
   529  647          37 GOC    FDEV40 ( 652) YES, IT IS O.K.
   530  650           1 GOSUB  SF51          SET FLAGS 55 & 21
   530  651           0                       *ILCAS&CTL:  CS0, @1345
   531  652 FDEV40    1 GOSUB  FNSTSA        READ THE STATUS
   531  653           0                       *ILCAS&CTL:  CS0, @0703
   532  654        1114 ?S9=1                PTR OR CASSETTE FOUND ?
   533  655          77 GOC    FDEV60 ( 664) NO, KEEP LOOKING
   534  656 RTNP+2  660 C=STK                YES, POP RTN ADDR TO C
   535  657        1072 C=C+1  M             INCREMENT RETURN ADDRESS
   536  660         740 GOTOC                GOTO RETURN ADDRESS + 1
   537  661 FDEV50 1670 C=REGN 14            IN MANUAL MODE ?
   538  662         574 RCR    6             C[S] = FLAGS 32-35
   539  663         776 C=C+C  S             FLAG 32 = MANUAL IL I/O
   540  664 FDEV60    1 GOLNC  UNT           NOT IN MANUAL MODE
   540  665           2                       *ILCAS&CTL,  CS0, @0254
   541  666        1536 ? A#0  S             LOOKING FOR PRINTER ?
   542  667        1523 GONC   FDEV36 ( 641) YES, SEE IF DEVICE EXISTS
   543  670        1623 GOTO   FDEV40 ( 652) NO, CHECK DEVICE STATUS
*****************************************************************
* RTPACH - PATCH FOR "READ TYPE" (RDTYPE) ROUTINE               *
*****************************************************************
   547                 ENTRY  RTPACH
   548  671 RTPACH  244 C=HPIL 2             READ STATUS BYTE FROM R2
   548  672         272                      (INSERTED BY ASSEMBLER)
   548  673         203                      (INSERTED BY ASSEMBLER)
   549  674        1200 HPIL=C 2             ECHO IT
   550  675           1 GOLONG RTPHRT        RETURN TO "RDTYPE" ROUTINE
   550  676           2                       *ILCAS&CTL:  CS0, @0512
   551                 FILLTO @677
        677        0000 NOP
*****************************************************************
* FNSTS  - FETCH NEW DEVICE STATUS                              *
* CSSTAS - SAME AS FNSTS EXCEPT ONLY READ ONE BYTE              *
*                                                               *
* ASSUMES: R5R/W = DEVICE LOOP ADDRESS                          *
*                                                               *
* INPUT:   IF S0 = 1 READ TWO BYTES OF STATUS                   *
*          IF S0 = 0 READ ONE BYTE OF STATUS                    *
* USES:    A, C, NO PT, +0 SUBROUTINE LEVELS                    *
*                                                               *
```

```
* OUTPUT:  A. FNSTS -                                                  *
*             1. ORIGINAL ST[7:0] IN C[1:0]                           *
*             2. 1ST BYTE OF STATUS IN S[7:0]                         *
*             3. 2ND BYTE OF STATUS IN C[13:12]                       *
*             4. BIT 3 OF 2ND STATUS BYTE IS SAVED IN BIT 3 OF R3R/W  *
*             5. ADDRESS THE DEVICE AS A LISTENER                     *
*             6. ADDRESS SELF AS A TALKER                             *
*          B. CSSTAS -                                                *
*             1. ORIGINAL S[7:0] IN C[1:0]                            *
*             2. STATUS BYTE IN S[7:0]                                *
*             3. ADDRESS THE DEVICE AS A LISTENER                     *
*             4. ADDRESS SELF AS A TALKER                             *
************************************************************************
   575                  ENTRY  FNSTS
   576                  ENTRY  FNSTSA
   577                  ENTRY  CSSTAS
*
   579  700 CSSTAS 1604 S0=    0             READ ONE BYTE OF STATUS
   580  701          23 GOTO   FNSTSA ( 703) CONTINUE READING DATA
   581  702 FNSTS  1610 S0=    1             READ TWO BYTES OF STATUS
   582  703 FNSTSA   16 A=0                  CLEAR THE "A" REGISTER
   583  704        1630 C=ST                 C[1:0] = STATUS FLAGS
   584  705        1114 ?S9=1                ERROR SO FAR ?
   585  706        1540 RTN C                YES, DON'T EVEN TRY
   586  707          44 HPL=CH 0             WRITE STATUS REGISTER
   587  710        1501 CH=    @320          SAY I'M SC, CA, LA
   588  711         144 HPL=CH 1             WRITE CONTROL INT REGISTER
   589  712        1005 CH=    @201          COMMAND CLASS (BIT 7)
* TEST IF TALKING TO A DUMB T.V.
   591  713         644 C=HPIL 6             GET DEVICE NUMBER
   591  714         672                      (INSERTED BY ASSEMBLER)
   591  715         603                      (INSERTED BY ASSEMBLER)
   592  716        1166 C=C-1  XS            VIDEO INTERFACE C[X]=FFF
   593  717        1046 C=C+1  X             TALKING TO A T.V. ?
   594  720         263 GONC   FSP10  ( 746) NO, IGNORE NEXT FEW LINES
   595  721        1160 DADD=C               IF USER FLG15 - TRACE MODE
   596  722         460 LDI                  IF USER FLG16 SET - NORMAL
   597  723          17 CON    @017           MODE TO TRACE MODE
   598  724         416 A=C                  A[1:0] = 0F HEX
   599  725        1670 C=REGN 14            LOAD FLAGS REGISTER
   600  726        1174 RCR    9             C[1:0] = FLAGS 12-19
   601  727        1730 CST EX               C[1:0] TO STATUS FLAGS
   602  730         114 ?S4=1                TRACE MODE ? (FLAG 15 SET)
   603  731          23 GONC   FSP04  ( 733) NO, SKIP NEXT LINE
   604  732         566 A=A+1  XS            A[XS] = 1
   605  733 FSP04    14 ?S3=1                NORMAL MODE ?(FLAG 16 SET)
   606  734          33 GONC   FSP06  ( 737) NO, SKIP NEXT 2 LINES
   607  735         566 A=A+1  XS            A[XS] = 2
   608  736         566 A=A+1  XS            A[XS] = 3
   609  737 FSP06  1730 CST EX               RESTORE STATUS FLAGS
   610  740         256 AC EX                C[XS] = 1 OR 3, C[X] = 0F
   611  741        1474 RCR    1             C[X] = 10 OR 30, C[S] = F
   612  742        1700 HPIL=C 7             R7R/W = 10 OR 30
   613  743        1074 RCR    2             C[S]=1 OR 3, C[M]=0F000...
   614  744         416 A=C                  A[S]=1 OR 3, A[M]=0F000...
   615  745         323 GOTO   FSPEX4 ( 777) WRITE STATUS/CONTROL/ETC.
   616  746 FSP10   544 C=HPIL 5             GET DEVICE LOOP ADDRESS
   616  747         572                      (INSERTED BY ASSEMBLER)
   616  750         503                      (INSERTED BY ASSEMBLER)
   617  751        1730 CST EX               C[1:0] TO STATUS FLAGS
```

```
618  752          510 S6=    1              MAKE IT A TALKER ADDRESS
619  753         1730 CST EX                 RESTORE C[1:0], STAT FLAGS
620  754         1200 HPIL=C 2               AND MAKE DEVICE A TALKER
621  755 FSP30    354 ORAV?                  READY TO SEND NEXT CMD ?
622  756          47 GOC    FSP45  ( 762)    YES, GO SEND NEXT COMMAND
623  757         1046 C=C+1  X                TIME OUT ?
624  760          137 GOC    FSPER  ( 773)    YES, INDICATE AN ERROR
625  761         1743 GOTO   FSP30  ( 755)    NOT YET, KEEP LOOPING
626  762 FSP45    144 HPL=CH 1                WRITE CONTROL INT REGISTER
627  763         1205 CH=    @241            READY CLASS (BITS 7 & 5)
628  764          244 HPL=CH 2                WRITE DATA BITS REGISTER
629  765          605 CH=    @141            SEND SST (SEND STATUS)
630  766 FSP48    106 C=0    X               INITIALIZE TIMER
631  767 FSP50    454 FRAV?                  READY TO SEND NEXT CMD ?
632  770          277 GOC    FSP60  (1017)    YES, GO READ NEXT BYTE
633  771         1046 C=C+1  X                TIME OUT ?
634  772         1753 GONC   FSP50  ( 767)    NOT YET
635  773 FSPER   1110 S9=    1               SET ERROR FLAG
636  774         1073 GOTO   FNSTSA ( 703)    GO BACK TO TOP AND RETURN
637  775 FSPEX   1614 ?S0=1                  PRINTER (READ 2 BYTES) ?
638  776          553 GONC   FSP85  (1053)    NO, READ ONLY ONE BYTE
639  777 FSPEX4    44 HPL=CH 0                WRITE STATUS REGISTER
640 1000         1611 CH=    @342            SAY I'M SC, CA, TA
641 1001          144 HPL=CH 1                WRITE CONTROL INT REGISTER
642 1002         1005 CH=    @201            COMMAND CLASS (BIT 7)
643 1003          544 C=HPIL 5                GET DEVICE LOOP ADDRESS
643 1004          572                        (INSERTED BY ASSEMBLER)
643 1005          503                        (INSERTED BY ASSEMBLER)
644 1006         1730 CST EX                 C[1:0] TO STATUS FLAGS
645 1007          210 S5=    1               MAKE IT LISTENER ADDRESS
646 1010         1730 CST EX                 RESTORE C[1:0], STAT FLAGS
647 1011         1200 HPIL=C 2               ADDRESS DEVICE AS LISTENER
648 1012 FSPEX6   354 ORAV?                  READY FOR THE NEXT FRAME ?
649 1013          367 GOC    FSP80  (1051)    YES, GET THE NEXT FRAME
650 1014         1046 C=C+1  X                TIME OUT ?
651 1015         1567 GOC    FSPER  ( 773)    YES, INDICATE AN ERROR
652 1016         1743 GOTO   FSPEX6 (1012)    NOT YET
653 1017 FSP60    244 C=HPIL 2                READ ONE BYTE
653 1020          272                        (INSERTED BY ASSEMBLER)
653 1021          203                        (INSERTED BY ASSEMBLER)
654 1022         1200 HPIL=C 2               RETRANSMIT LAST DATA FRAME
655 1023         1614 ?S0=1                  PRINTER (READ 2 BYTES) ?
656 1024          123 GONC   FSP68  (1036)    NO, DONE READING 1 BYTE
657 1025         1536 ? A#0  S                HAS TO READ ANOTHER BYTE ?
658 1026           57 GOC    FSP65  (1033)    NO, SAVE STAT BYTE 2 BIT 3
659 1027          576 A=A+1  S                REMEMBER HAVING READ 1 BYTE
660 1030          406 A=C    X                SAVE FIRST BYTE IN A[X]
661 1031         1700 HPIL=C 7                SAVE 1ST BYTE OF STS IN R7
662 1032         1343 GOTO   FSP48  ( 766)    GOTO READ NEXT BYTE
663 1033 FSP65   1600 HPIL=C 6                SAVE 2ND BYTE BIT 3 TO R6
664 1034         1074 RCR    2               C[13:12] = 2ND BYTE
665 1035          246 AC EX  X                C[1:0] = 1ST BYTE
666 1036 FSP68   1074 RCR    2               C[13:12] = 1ST & 2ND BYTE
667 1037          416 A=C                    SAVE WHOLE THING IN A TEMP
668 1040 FSP70    454 FRAV?                  ETO OR ETE COME IN YET ?
669 1041           47 GOC    FSP75  (1045)    YES, DEAL WITH ETO OR ETE
670 1042         1046 C=C+1  X                TIME OUT YET ?
671 1043         1307 GOC    FSPER  ( 773)    YES, INDICATE AN ERROR
672 1044         1743 GOTO   FSP70  (1040)    NOT TIME OUT YET
673 1045 FSP75    244 C=HPIL 2                READ ETO/ETE
```

```
   673 1046         272                        (INSERTED BY ASSEMBLER)
   673 1047         203                        (INSERTED BY ASSEMBLER)
   674 1050        1253 GOTO   FSPEX  ( 775) GET 2ND BYTE IF PRINTER
   675 1051 FSP80  1154 FRNS?                 FRAME RETURN NOT AS SENT ?
   676 1052        1217 GOC    FSPER  ( 773) YES, INDICATE AN ERROR
   677 1053 FSP85   256 AC EX                 SAVE ETO/ETE TO "A" REG
   678 1053          53 GOTO   FS100  (1061) PUT C[13:12] INTO ST[7:0]
*
*************************************************************************
* OOPCHK (OUT-OF-PAPER CHECK) - CHECKS FOR OUT-OF-PAPER STATUS AND      *
* PUTS SECOND STATUS BYTE INTO ST[7:0]                                  *
*                                                                       *
* ON ENTRY ASSUMES FIRST BYTE OF PRINTER STATUS IS IN ST[7:0]           *
* IF OOPS THEN SETS S9.                                                 *
* ALWAYS PUTS WHAT WAS IN C[13:12] ON ENTRY INTO ST[7:0]                *
* AND PUTS FIRST PRINTER STATUS BYTE TO C[3:2].                         *
* C[1:0] IS PRESERVED.                                                  *
*                                                                       *
*************************************************************************
   691                  ENTRY  FS100
   692                  ENTRY  OOPCHK
   693 1055 OOPCHK   14 ?S3=1                 OUT OF PAPER (OOPS) ?
   694 1056          23 GONC   OOP10  (1060) NO, CONTINUE PROCESSING
   695 1057        1110 S9=    1              YES, SET ERROR FLAG
   696 1060 OOP10  1730 CST EX                C[1:0]=PRINTER STATUS FLAGS
   697 1061 FS100  1574 RCR    12             C[3:2]=PRINTER STATUS FLAGS
   698 1062        1730 CST EX                STAT FLAGS = PREV. C[13:12]
   699 1063        1740 RTN                   END OF OUT OF PAPER CHECK
*
*************************************************************************
* OUTA   - SEND ALPHA REG OUT THROUGH PIL                               *
* AOUTFL - SAME AS AOUT1 EXCEPT WILL PICK UP THE NAME FROM THE          *
*          ADDRESS SAVED IN REG.8[13:10]                                *
* AOUTIN - PUT THE BEGINNING ADDR OF ALPHA REGISTER IN REG.8[13:10]     *
*                                                                       *
* THE FILE NAME WILL BE TERMINATED BY EITHER A COMMA OR END OF ALPHA    *
* REGISTER.  A FILE NAME HAS TO BE EXACTLY 7 CHARACTERS.  IT WILL BE    *
* TRUNCATED OR FILLED WITH TRAILING BLANKS IF IT IS LONGER OR SHORTER   *
* THAN 7 CHARACTERS.  THE ADDRESS OF A DELIMINATOR WILL BE SAVED IN     *
* REG.8[13:10] EVERY TIME FOR LATER USE                                 *
*                                                                       *
* USES :   A, B[3:0], C, S[7:4], +1 SUBROUTINE LEVEL                    *
*                                                                       *
* STATUS USED:                                                          *
*   S7 :   IF S7=0, SEND OUT ENTIRE ALPHA REG (FOR AOUT FUNCTION)       *
*          IF S7=1, GET FILE NAME FROM ALPHA REGISTER                   *
*   S6 :   SET TO "0" AT BEGINNING, WILL BE SET TO "1" FOR ANY          *
*          NON-NULL CHARACTER IN THE ALPHA REGISTER                     *
*   S5 :   SET TO "0" AT BEGINNING, WILL BE SET TO "1" IF A COMMA       *
*          IS ENCOUNTERED                                               *
*   S3 :   IF S3=1, FILE WILL BE SHIFTED INTO M FROM LEFT END           *
*          IF S3=0, FILE WILL BE SHIFTED INTO M FROM RIGHT END          *
*   S2 :   IF S2=1, LAST CHAR ADDRESS WILL BE SAVED IN REG.8[13:10]     *
*          IF S2=0, DON'T CHANGE REG.8 AT ALL                           *
*                                                                       *
*************************************************************************
   728                  ENTRY  OUTA
   729                  ENTRY  AOUT1
   730                  ENTRY  AOUTIN
   731                  ENTRY  AOUTFL
```

```
*
   733 1064 AOUT1     1 GOSUB  AOUTIN         SET STARTING ADDRESS
   733 1065           0                       *ILCAS&CTL:  CS0, @1266
   734 1066 AOUTFL 1210 S7=    1              SET FLAG TO GET FILE NAME
   735 1067         504 S6=    0              CLEAR NULL STRING FLAG
   736 1070         204 S5=    0              RESET COMMA FLAG
   737 1071        1070 C=REGN 8              ALPHA REG SCRATCH, 22-24
   738 1072          34 PT=    3              POINT AT DIGIT 3
   739 1073         374 RCR    10             C[3:0] = C[13:10]
   740 1074         352 BC EX  WPT            PUT LAST ADDR IN "B"
   741 1075        1334 PT=    13             POINT AT C MANTISSA SIGN
   742 1076         720 LC     7              WRITE CHARACTER COUNTER
   743 1077         276 AC EX  S              CHAR COUNTER IN A[13]
   744 1100          34 PT=    3              REPOSITION TO DIGIT 3
   745 1101         633 GOTO   AOUT20 (1164) GET ALPHA ADDRESS, ETC.
*
   747 1102         201 CON    @201           A
   748 1103          24 CON    @24            T
   749 1104          25 CON    @25            U
   750 1105          17 CON    @17            O
   751 1106 OUTA      1 GOSUB  SCHDEV         SEARCH G.P. INTERFACE MOD
   751 1107           0                       *ILCAS&CTL:  CS0, @1335
   752 1110           1 GOSUB  LISTEN         ADDRESS DEVICE AS LISTENER
   752 1111           0                       *ILCAS&CTL:  CS0, @0335
   753 1112        1704 CLR ST                CLEAR STATUS FLAGS
   754 1113           1 GOSUB  AOUTIN         SET STARTING ADDRESS
   754 1114           0                       *ILCAS&CTL:  CS0, @1266
   755 1115 AOUT10    1 GOSUB  INCADA         INCREMENT ADDRESS IN RAM
   755 1116           0                       *MAINFRAME: CN10, @0726
   756 1117           1 GOSUB  GTBYTA         GET NEXT BYTE
   756 1120           0                       *MAINFRAME: CN10, @0673
   757 1121         212 B=A    WPT            ADDRESS TO REGISTER B
   758 1122        1434 PT=    1              POINT AT DIGIT 1 OF "C"
   759 1123        1352 ? C#0  WPT            IS IT A LEADING BLANK ?
   760 1124         403 GONC   AOUT20 (1164) YES, IT IS A LEADING BLANK
   761 1125         510 S6=    1              SET NON-NULL STRING FLAG
   762 1126        1214 ?S7=1                 OUTPUTTING FILE NAME ?
   763 1127         333 GONC   AOUT18 (1162) NO, SEND IT
   764 1130         412 A=C    WPT            YES, CHECK FOR COMMA
   765 1131         460 LDI                   LOAD LOW 12 BITS OF C WITH
   766 1132          54 CON    @54            @54 = ASCII CODE FOR COMMA
   767 1133        1552 ? A#C  WPT            IS CHARACTER A COMMA ?
   768 1134          37 GOC    AOUT12 (1137) NO, SOME OTHER CHARACTER
   769 1135         210 S5=    1              INDICATE COMMA ENCOUNTERED
   770 1136         663 GOTO   AOUT32 (1224) MOVE ON TO NEXT CHARACTER
   771 1137 AOUT12 1536 ? A#0  S              7 CHARACTERS YET ?
   772 1140         243 GONC   AOUT20 (1164) YES, DONE WITH STRING
   773 1141         252 AC EX  WPT            SWAP A[1:0] WITH C[1:0]
   774 1142 AOUT14  676 A=A-1  S              DEC. CHAR COUNT
   775 1143 AOUT15 1434 PT=    1              POINT AT LOWEST BYTE
   776 1144         412 A=C    WPT            A[1:0] = BYTE AT C[1:0]
   777 1145         630 C=M                   GET CONTENTS OF "M" REG
   778 1146          14 ?S3=1                 SHIFT TO M FROM LEFT ?
   779 1147          43 GONC   AOUT16 (1153) NO, FROM RIGHT
   780 1150         252 AC EX  WPT            SHIFT CHAR TO M FROM LEFT
   781 1151        1074 RCR    2              ROTATE RIGHT ONE BYTE
   782 1152          63 GOTO   AOUT17 (1160) RESTORE M FROM C
   783 1153 AOUT16  256 AC EX                SHIFT CHAR TO M FROM RIGHT
   784 1154        1756 A SL                 SHIFT LEFT ONE DIGIT
   785 1155        1756 A SL                 SHIFT LEFT ANOTHER DIGIT
```

```
786 1156          412 A=C    WPT           A[1:0] = CHAR FROM C[1:0]
787 1157          256 AC EX                BRING STRING TO C FROM A
788 1160 AOUT17   530 M=C                  RESTORE M FROM C
789 1161           33 GOTO   AOUT20 (1164) CONTINUE PROCESSING STRING
790 1162 AOUT18     1 GOSUB  SDATA0        NO, SEND CHARACTER
790 1163           0                      *ILCAS&CTL:  CS0, @0446
791 1164 AOUT20    34 PT=    3            POINT AT LOWEST 2 BYTES
792 1165          152 AB EX  WPT           ADDRESS BACK TO A
793 1166          460 LDI                  LOAD LOW 12 BITS OF C WITH
794 1167            5 CON2   0      5      C[1:0]=05 HEX
795 1170          102 C=0    PT            CLEAR C[3] TO ZERO
796 1171         1552 ? A#C  WPT           END OF "A" REGISTER ?
797 1172         1237 GOC    AOUT10 (1115) NOT YET
798 1173          212 B=A    WPT           ADDRESS TO B
799 1174          514 ?S6=1                NULL STRING ?
800 1175           67 GOC    AOUT30 (1203) NO, STRING IS NON-NULL
801 1176         1214 ?S7=1                OUTPUTTING FILE NAME ?
802 1177          233 GONC   AOUT31 (1222) NO, SEND ALPHA REGISTER
803 1200           14 ?S3=1                SHIFT FROM LEFT ?
804 1201          553 GONC   FLNMER (1256) NO, FILENAME ERROR
805 1202         1740 RTN                  YES, RETURN NULL STRING
806 1203 AOUT30  1214 ?S7=1                OUTPUTTING FILE NAME ?
807 1204          207 GOC    AOUT32 (1224) YES, SEND FILE NAME
808 1205         1670 C=REGN 14            READ FLAGS REGISTER
809 1206          374 RCR    10            C[S] = FLAGS 16-19
810 1207          776 C=C+C  S             FLAG 16 SENT TO CARRY
811 1210          776 C=C+C  S             FLAG 17 TO CARRY - CHECK IT
812 1211          117 GOC    AOUT31 (1222) IF SET, SUPPRESS CR, LF
813 1212          460 LDI                  LOAD LOW 12 BITS OF C WITH
814 1213           15 CON    13            13 = ASCII CARRIAGE RETURN
815 1214            1 GOSUB  SDATA         SEND "CR" (CARRIAGE RETURN)
815 1215            0                      *ILCAS&CTL:  CS0, @0450
816 1216          460 LDI                  LOAD LOW 12 BITS OF C WITH
817 1217           12 CON    10            10 = ASCII LINE FEED
818 1220            1 GOSUB  SDATA         SEND "LF" (LINE FEED)
818 1221            0                      *ILCAS&CTL:  CS0, @0450
819 1222 AOUT31     1 GOLONG UNLCHK        UNLISTEN & CHECK FOR ERROR
819 1223            2                      *ILCAS&CTL:  CS0, @1466
820 1224 AOUT32  1536 ? A#0  S             7 CHARACTERS YET ?
821 1225          153 GONC   AOUT40 (1242) YES, DONE WITH STRING
822 1226          630 C=M                  GET CONTENTS OF "M" REG
823 1227           14 ?S3=1                SHIFT TO M FROM LEFT ?
824 1230           53 GONC   AOUT35 (1235) NO, FROM RIGHT
825 1231         1074 RCR    2             ROTATE RIGHT ONE BYTE
826 1232 AOUT33   530 M=C                  RESTORE M FROM C
827 1233 AOUT34   676 A=A-1  S             DECREMENT CHARACTER COUNT
828                    LEGAL               (CLEAR THE CARRY FLAG)
829 1234         1703 GOTO   AOUT32 (1224) PROCESS NEXT CHARACTER
830 1235 AOUT35  1574 RCR    12            ROTATE BLANK TO M F/RIGHT
831 1236         1434 PT=    1             POINT TO LOWEST BYTE
832 1237          220 LC     2             C[1] = 2
833 1240           20 LC     0             C[1:0] = 20H (ASCII SPACE)
834 1241         1713 GOTO   AOUT33 (1232) DEC CHAR COUNT, GET NEXT
835 1242 AOUT40  1014 ?S2=1                NEED TO SAVE ADDRESS ?
836 1243          113 GONC   AOUT55 (1254) NO, ERROR CHECK & RETURN
837 1244           34 PT=    3             POINT AT LOWEST 2 BYTES
838 1245          152 AB EX  WPT           A[3:0] = SAVED ALPHA ADDR
839 1246 AOUT50  1070 C=REGN 8             ALPHA REG SCRATCH, 22-24
840 1247          374 RCR    10            C[3:0] = ALPHA SCRATCH
841 1250          252 C=A    WPT           C[3:0] = ALPHA START ADDR
```

```
   841 1251          412                        (INSERTED BY ASSEMBLER)
   842 1252          174 RCR     4              C[13:10] = ALPHA START
   843 1253         1050 REGN=C  8              STO ADDR IN REG.8[13:10]
   844 1254 AOUT55     1 GOLONG  PLERCK          ERROR CHECK AND RETURN
   844 1255            2                        *ILCAS&CTL:  CS1, @1747
*
   846                   ENTRY   FLNMER
   847 1256 FLNMER     1 GOSUB   PLEREX          PIL ERROR EXIT : NAME
   847 1257            0                        *ILCAS&CTL:  CS1, @1663
   848 1260           16 CON     @16            N
   849 1261            1 CON     @01            A
   850 1262           15 CON     @15            M
   851 1263         1005 CON     @1005          E
   852 1264 DSPERJ     1 GOLONG  DSPERR          DISPLAY MESSAGE: ERR
   852 1265            2                        *ILCAS&CTL:  CS3, @0405
*
   854 1266 AOUTIN   116 C=0                    CLEAR ACCUMULATOR
   855 1267           34 PT=     3              POINT AT LOWEST 2 BYTES
   856 1270          620 LC      6              C[3:0] = 6000 HEX = BYTE 3
   857 1271         1634 PT=     0              POINT AT LOWEST DIGIT
   858 1272         1020 LC      8              C[3:0] = 6008 HEX = REG. 8
   859 1273           34 PT=     3              POINT AT LOWEST 2 BYTES
   860 1274          412 A=C     WPT            A[3:0] = ALPHA START ADDR
   861 1275         1104 S9=     0              CLEAR ERROR FLAG
   862 1276         1503 GOTO    AOUT50 (1246)  REG.8[13:10] = ALPHA START
*
*
*************************************************************************
* SELECT - SELECT DEVICE NUMBER                                         *
*          GET INTEGER PART IN X-REG AS DEVICE NUMBER,                  *
*          CONVERT IT TO BINARY AND STORE IT IN R4R/W.                  *
*          THE NUMBER MUST BE BETWEEN 1 AND 30, INCLUSIVE.              *
*************************************************************************
*
*
   873                   ENTRY   SELECT
   874                   ENTRY   CHKLAD
   875                   ENTRY   LADERM
*
   877 1277          224 CON     @224           T
   878 1300            3 CON     @03            C
   879 1301            5 CON     @05            E
   880 1302           14 CON     @14            L
   881 1303            5 CON     @05            E
   882 1304           23 CON     @23            S
   883 1305 SELECT     1 GOSUB   CONV3D         CONVERT X-REG TO BINARY
   883 1306            0                        *ILCAS&CTL:  CS3, @1465
   884 1307          410 S8=     1              SET TO DISPLAY ERR MESSAGE
   885 1310 CHKLAD  1346 ? C#0   X              LISTENER ADDRESS = 0 ?
   886 1311          113 GONC    LADER  (1322)  YES, THIS IS AN ERROR
   887 1312          406 A=C     X              SAVE LISTENER ADDR TO A[X]
   888 1313          460 LDI                    LOAD LOW 12 BITS OF C WITH
   889 1314           37 CON     31             ONE MORE THAN MAX LISTENER
   890 1315         1406 ? A<C   X              LAD < 31 ?
   891 1316           43 GONC    LADER  (1322)  NO, THIS IS AN ERROR
   892 1317          246 AC EX   X              RESTORE LISTENER ADDRESS
   893 1320 SELD10  1400 HPIL=C 4              WRITE LISTEN ADDR TO R4R/W
   894 1321         1740 RTN                   END OF SELECT ROUTINE
   895 1322 LADER    414 ?S8=1                 DISPLAY ERROR MESSAGE ?
   896 1323           47 GOC     LADERM (1327)  YES, GO TO ERROR MESSAGE
```

```
   897 1324         106 C=0    X             NO, SET DEFAULT DEVICE #
   898 1325        1046 C=C+1  X             SET SELECTED DEV # TO 1
   899                  LEGAL                 (CLEAR THE CARRY FLAG)
   900 1326        1723 GOTO   SELD10 (1320) DEV # TO LOOP ADDRESS REG
   901 1327 LADERM    1 GOSUB  PLEREX        PIL ERROR EXIT : ADR
   901 1330           0                      *ILCAS&CTL:  CS1, @1663
   902 1331           1 CON    @01           A
   903 1332           4 CON    @04           D
   904 1333        1022 CON    @1022         R
   905 1334        1303 GOTO   DSPERJ (1264) DISPLAY MESSAGE : ERR
*
*********************************************************************
* SCHDEV - GET SELECTED DEVICE NUMBER                               *
*                                                                   *
* USES:    A[X], C, +1 SUBROUTINE LEVEL                             *
* OUTPUT:  R5R/W = DEVICE NUMBER                                    *
*********************************************************************
   913                  ENTRY  SCHDEV
*
   915 1335 SCHDEV    1 GOSUB  PILEN         ENABLE PIL CHIP
   915 1336           0                      *ILCAS&CTL:  CS0, @0143
   916 1337           1 GOSUB  PLERCK        CHECK FOR ANY ERRORS
   916 1340           0                      *ILCAS&CTL:  CS1, @1747
   917 1341           1 GOLONG GETDEV        GET START SEARCHING DEV #
   917 1342           2                      *ILCAS&CTL:  CS0, @0520
*
*********************************************************************
* SF5521 - SET FLAGS 55 AND 21                                      *
*                                                                   *
* USES:    C, ST, AND 1 ADDITIONAL SUBROUTINE LEVEL                 *
* INPUT:   NOTHING                                                  *
* OUTPUT:  SS 0 UP, CHIP 0 ENABLED, C = REG 14                      *
* ASSUMES: NOTHING                                                  *
*********************************************************************
   927                  ENTRY  SF5521
   928                  ENTRY  SF51
   929 1343 SF5521    1 GOSUB  LDSST0        GET FLAGS 48-55 TO STATUS
   929 1344           0                      *MAINFRAME:  CN1, @1627
   930 1345 SF51   1610 S0=    1             SET PRINTER EXISTENCE FLAG
   931 1346        1630 C=ST                 USER FLAGS 48-55 TO C[1:0]
   932 1347         474 RCR    8             C[1:0] = FLAGS 16-23
   933 1350        1730 CST EX               MOVE FLAGS 16-23 TO STATUS
   934 1351        1010 S2=    1             SET PRINTER CONTROL FLAG
   935 1352        1730 CST EX               (FLAG 21 = ENABLE PRINTER)
   936 1353         574 RCR    6             ROTATE FLAGS BACK TO NORMAL
   937 1354 SFRTN  1650 REGN=C 14            PUT FLAGS BACK TO REG. 14
   938 1355         363 GOTO   SETIDY (1413) SET AUTO IDENTIFY SOURCING
*
*********************************************************************
* MANIO -- SET   UNFRIENDLY FLAG - USER FLAG 32                     *
* AUTOIO - RESET UNFRIENDLY FLAG - USER FLAG 32                     *
* RSIDY -- SET AUTO IDENTIFY BIT - PARALLEL POLL REGISTER 3, BIT 6  *
*********************************************************************
*
   946                  ENTRY  MANIO
   947                  ENTRY  AUTOIO
   948                  ENTRY  RSIDY
*
   950 1356         217 CON    @217          O
   951 1357          11 CON    @11           I
```

```
   952 1360           16 CON    @16           N
   953 1361            1 CON    @01           A
   954 1362           15 CON    @15           M
   955 1363 MANIO    410 S8=    1             S8 IS SET FOR MANUAL MODE
   956 1364          103 GOTO   FLAG32 (1374) SET UNFRIENDLY FLAG 32
   957 1365          217 CON    @217          O
   958 1366           11 CON    @11           I
   959 1367           17 CON    @17           O
   960 1370           24 CON    @24           T
   961 1371           25 CON    @25           U
   962 1372            1 CON    @01           A
   963 1373 AUTOIO   404 S8=    0             S8 IS RESET FOR AUTO MODE
   964 1374 FLAG32  1670 C=REGN 14            READ FLAGS REGISTER
   965 1375          274 RCR    5             C[1:0] = FLAGS 28-35
   966 1376         1730 CST EX               PUT FLAGS 28-35 TO STATUS
   967 1377            4 S3=    0             CLEAR FLAG 32 FOR AUTO
   968 1400          414 ?S8=1                IS THIS MANUAL I/O MODE ?
   969 1401           23 GONC   *+2    (1403) NO, AUTOMATIC I/O MODE
   970 1402           10 S3=    1             SET FLAG 32 FOR MANUAL
   971 1403         1730 CST EX               RESTORE FLAGS & STATUS
   972 1404         1174 RCR    9             ROTATE FLAGS BACK TO NORMAL
   973 1405         1650 REGN=C 14            PUT FLAGS BACK TO REG. 14
   974 1406          414 ?S8=1                GO INTO AUTO MODE ?
   975 1407         1540 RTN C                NO, WE'RE DONE HERE
   976 1410 RSIDY      1 GOSUB  FNDPTR        SEE IF PRINTER PLUGGED IN
   976 1411            0                      *ILCAS&CTL:  CS0, @0575
   977 1412         1740 RTN                  P+1 - NO PRINTER FOUND
   978 1413 SETIDY   344 HPL=CH 3             P+2 - WRITE PAR POLL REG 3
   979 1414          401 CH=    @100          SET AUTO IDENTIFY BIT
   980 1415            1 GOLONG UNL           SEND UNLISTEN COMMAND
   980 1416            2                      *ILCAS&CTL:  CS0, @0257
*
*************************************************************************
* REMOTE - SEND COMMAND "REMOTE ENABLE"                                 *
* LOCAL  - SEND COMMAND "GO TO LOCAL"                                   *
* TRIGER - SEND COMMAND "GROUP EXECUTE TRIGGER" (USER FUNC: "TRIGGER")  *
* NOTES :  1. ALL ABOVE FUNCTIONS WILL SEARCH FOR G.P.I.O. IF IN AUTO   *
*             MODE, OTHERWISE WILL USE SELECTED DEVICE.                 *
*          2. ALL ABOVE FUNCTIONS WILL ADDRESS THE SELECTED DEVICE AS   *
*             A LISTENER BEFORE SENDING THE COMMAND AND THEN SEND AN    *
*             UNLISTEN COMMAND AFTERWARD.                               *
* UNLCHK - SEND UNLISTEN COMMAND, THEN CHECK FOR ERRORS                 *
*************************************************************************
*
   994                ENTRY  REMOTE
   995                ENTRY  LOCAL
   996                ENTRY  TRIGER
   997                ENTRY  UNLCHK
*
*
  1000 1417          222 CON    @222          R
  1001 1420            5 CON    @05           E
  1002 1421            7 CON    @07           G
  1003 1422            7 CON    @07           G
  1004 1423           11 CON    @11           I
  1005 1424           22 CON    @22           R
  1006 1425           24 CON    @24           T
  1007 1426 TRIGER   460 LDI                   SEND GROUP EXECUTE TRIGGER
  1008 1427           10 CON    @010          @010 = "GET" COMMAND
  1009 1430          263 GOTO   SNCMD  (1456) SEND COMMAND TO DEVICE
```

```
*
  1011 1431          214 CON    @214           L
  1012 1432            1 CON    @01            A
  1013 1433            3 CON    @03            C
  1014 1434           17 CON    @17            O
  1015 1435           14 CON    @14            L
  1016 1436 LOCAL    460 LDI                   SEND GO TO LOCAL COMMAND
  1017 1437            1 CON    @001           @001 = "GTL" COMMAND
  1018 1440          163 GOTO   SNCMD  (1456)  SEND COMMAND TO DEVICE
*
  1020 1441          205 CON    @205           E
  1021 1442           24 CON    @24            T
  1022 1443           17 CON    @17            O
  1023 1444           15 CON    @15            M
  1024 1445            5 CON    @05            E
  1025 1446           22 CON    @22            R
  1026 1447 REMOTE     1 GOSUB  SCHDEV         SEARCH FOR THE DEVICE
  1026 1450            0                       *ILCAS&CTL:  CS0, @1335
  1027 1451          460 LDI                   SEND REMOTE ENABLE
  1028 1452          222 CON    @222           @222 = "REN" COMMAND
  1029 1453            1 GOSUB  SCMD           SEND COMMAND TO DEVICE
  1029 1454            0                       *ILCAS&CTL:  CS0, @0272
  1030 1455          106 C=0    X              CLEAR C[2:0] (NO COMMAND)
*
  1032 1456 SNCMD    346 CB EX  X              SAVE THE COMMAND IN B[X]
  1033 1457            1 GOSUB  SCHDEV         SEARCH FOR THE DEVICE
  1033 1460            0                       *ILCAS&CTL:  CS0, @1335
  1034 1461            1 GOSUB  LISTEN         ADDRESS DEVICE AS LISTENER
  1034 1462            0                       *ILCAS&CTL:  CS0, @0335
  1035 1463          306 C=B    X              RETRIEVE COMMAND FROM B[X]
  1036 1464            1 GOSUB  SCMD           SEND COMMAND TO DEVICE
  1036 1465            0                       *ILCAS&CTL:  CS0, @0272
  1037 1466 UNLCHK     1 GOSUB  UNL            SEND UNLISTEN COMMAND
  1037 1467            0                       *ILCAS&CTL:  CS0, @0257
  1038 1470            1 GOSUB  PLERCK         CHECK FOR ANY ERRORS
  1038 1471            0                       *ILCAS&CTL:  CS1, @1747
*
*************************************************************************
* LISTNR - ADDRESS ANY DEVICE AS A LISTENER (USER FUNCTION: "LISTEN")  *
* THE INTEGER PART OF X IS TAKEN AS THE DEVICE NUMBER (1<=NUMBER<=31)  *
*************************************************************************
*
  1045                  ENTRY  LISTNR
  1046 1472          216 CON    @216           N
  1047 1473            5 CON    @05            E
  1048 1474           24 CON    @24            T
  1049 1475           23 CON    @23            S
  1050 1476           11 CON    @11            I
  1051 1477           14 CON    @14            L
  1052 1500 LISTNR     1 GOSUB  CONV3D         CONVERT X-REG TO BINARY
  1052 1501            0                       *ILCAS&CTL:  CS3, @1465
  1053 1502          406 A=C    X              A[X] = BINARY VALUE OF X
  1054 1503          460 LDI                   LOAD LOW 12 BITS OF C WITH
  1055 1504           40 CON    32             ONE MORE THAN MAX DEVICE #
  1056 1505         1406 ? A<C  X              IS DEVICE NUMBER < 32 ?
  1057 1506 LISTER     1 GOLNC  LADERM         NO, DISPLAY "ADR ERR"
  1057 1507            2                       *ILCAS&CTL:  CS0, @1327
  1058 1510         1506 ? A#0  X              DOES DEV NUMBER EQUAL 0 ?
  1059 1511         1753 GONC   LISTER (1506) YES, INDICATE AN ERROR
  1060 1512          246 AC EX  X              C[X] = DEVICE NUMBER
```

```
  1061 1513         1500 HPIL=C 5              WRITE DEV # TO R5R/W
  1062 1514            1 GOSUB  PILEN          ENABLE PIL CHIP
  1062 1515            0                       *ILCAS&CTL:  CS0, @0143
  1063 1516            1 GOSUB  PLERCK         CHECK FOR ANY ERRORS
  1063 1517            0                       *ILCAS&CTL:  CS1, @1747
  1064 1520            1 GOLONG LISTEN         MAKE DEVICE A LISTENER
  1064 1521            2                       *ILCAS&CTL:  CS0, @0335
*
*************************************************************************
* INA  - INPUT AN ASCII STRING AND PUT IT TO THE ALPHA REGISTER         *
* INDX - INPUT AN ASCII STRING AND PUT IT TO X-REGISTER AS A DECIMAL    *
* NOTE:  USER FUNCTION NAME FOR INDX IS "IND"                           *
*************************************************************************
  1071                   ENTRY  INA
  1072                   ENTRY  INDX
  1073                   ENTRY  INAD2          ** ADDED ON JUNE 3, 1981
  1074                   ENTRY  INADRD
*
  1076 1522          204 CON    @204           D
  1077 1523           16 CON    @16            N
  1078 1524           11 CON    @11            I
  1079 1525 INDX     410 S8=    1              SET S8 FOR ASCII TO DECIMAL
  1080 1526          116 C=0                   INITIALIZE DIGIT ENTRY
  1081 1527         1156 C=C-1                 C = FFFFFFFFFFFFF HEX
  1082 1530          126 C=0    XS             C[X] = 0FF HEX
  1083 1531         1334 PT=    13             POINT AT HIGHEST DIGIT
  1084 1532         1220 LC     10             C = AFFFFFFFFF0FF
  1085 1533         1150 REGN=C 9              WRITE TO TEMP ALPHA SCRATCH
  1086 1534            1 GOSUB  STBT10         MOVE BITS TO SCRATCH REG. 8
  1086 1535            0                       *MAINFRAME: CN11, @1243
  1087 1536          634 PT=    11             POINT AT DIGIT 11 OF C
  1088 1537         1020 LC     8              SAY MANTISSA NON-ZERO
  1089 1540         1050 REGN=C 8              REWRITE SCRATCH AREA REG. 8
  1090 1541          123 GOTO   INAD   (1553) GET DEVICE ADDRESS, ETC.
*
  1092 1542          201 CON    @201           A
  1093 1543           16 CON    @16            N
  1094 1544           11 CON    @11            I
  1095 1545 INA      404 S8=    0              CLEAR S8 FOR ASCII TO ALPHA
  1096 1546          116 C=0                   CLEAR ACCUMULATOR
  1097 1547          460 LDI                   LOAD LOW 12 BITS OF C WITH
  1098 1550           27 CON    23             COUNT - 1 (24 ALPHA CHARS)
  1099 1551          674 RCR    11             MOVE COUNT TO LOW BYTE C[M]
  1100 1552          356 BC EX                 THEN PRESERVE IT IN B[M]
  1101 1553 INAD       1 GOSUB  SCHDEV         GET DEVICE ADDRESS
  1101 1554            0                       *ILCAS&CTL:  CS0, @1335
  1102 1555         1704 CLR ST                NOT FOR FIND ID
  1103 1556          414 ?S8=1                 ASCII TO DECIMAL (IND) ?
  1104 1557            1 GSUBNC CLA            NO, CLEAR ALPHA REGISTER
  1104 1560            0                       *MAINFRAME:  CN4, @0321
  1105 1561            1 GOSUB  TALKER         ADDRESS DEVICE AS A TALKER
  1105 1562            0                       *ILCAS&CTL:  CS0, @0262
  1106 1563 INAD2   1670 C=REGN 14             READ FLAGS REGISTER
  1107 1564          374 RCR    10             C[S] = FLAGS 16-19
  1108 1565          776 C=C+C  S              SHIFT FLAG 16 INTO CARRY
  1109 1566          776 C=C+C  S              SHIFT FLAG 17 INTO CARRY
  1110 1567           23 GONC   INAD00 (1571) IF FLAG 17 CLEAR, S2 = 0
  1111 1570         1010 S2=    1              FLAG 17 SET (IGNORE CR,LF)
  1112 1571 INAD00     1 GOSUB  NATNRD         SEND READY COMMAND "NAT"
  1112 1572            0                       *ILCAS&CTL:  CS0, @0352
```

```
1113 1573         1104 S9=     0              RESET ERR FLAG IF TIME OUT
1114 1574           33 GOTO    INADRD (1577)  READ A DATA FRAME, ETC.
1115 1575 INAECO   306 C=B     X              MOVE CHAR COUNT TO C[M]
1116 1576         1200 HPIL=C  2              RETRANSMIT LAST DATA FRAME
1117 1577 INADRD     1 GOSUB   RDDFRM         READ A DATA FRAME
1117 1600            0                        *ILCAS&CTL:  CS0, @0420
1118 1601         1114 ?S9=1                  IS IT A DATA FRAME ?
1119 1602            1 GOLC    INADEX         NO, MIGHT BE AN ETO
1119 1603            3                        *ILCAS&CTL:  CS0, @1716
1120 1604         1634 PT=     0              POINT AT LOWEST BYTE
1121 1605          130 G=C                    PUT LOWEST BYTE TO G
1122 1606          406 A=C     X              SAVE THE FRAME IN A[X]
1123 1607          206 B=A     X              SAVE THE FRAME IN B[X]
1124 1610          414 ?S8=1                  ASCII TO DECIMAL (IND) ?
1125 1611           37 GOC     CKCRLF (1614)  YES, CHECK IF CR/LF
1126 1612         1014 ?S2=1                  NO, FLAG 17 SET ?
1127 1613          157 GOC     INAD20 (1630)  YES, IGNORE CR/LF
1128 1614 CKCRLF   460 LDI                    LOAD LOW 12 BITS OF C WITH
1129 1615           15 CON     13             13 = ASCII CARRIAGE RETURN
1130 1616         1546 ? A#C   X              IS IT A CARRIAGE RETURN ?
1131 1617         1563 GONC    INAECO (1575)  YES, IGNORE IT
1132 1620          460 LDI                    LOAD LOW 12 BITS OF C WITH
1133 1621           12 CON     10             10 = ASCII LINE FEED
1134 1622         1546 ? A#C   X              IS IT A LINE FEED ?
1135 1623           57 GOC     INAD20 (1630)  NO, CONTINUE PROCESSING
1136 1624 INANRD   306 C=B     X              GET LAST DATA FRAME
1137 1625            1 GOSUB   NRDC           SEND NOT READY FOR DATA
1137 1626            0                        *ILCAS&CTL:  CS0, @0366
1138 1627          753 GOTO    INERCK (1724)  INPUT ERROR CHECK
1139 1630 INAD20   414 ?S8=1                  ASCII TO DECIMAL (IND) ?
1140 1631          227 GOC     IND10  (1653)  YES, FIND DIGITS, ETC.
1141 1632 INAD22    14 ?S3=1                  FIND ID ?
1142 1633          113 GONC    INAD25 (1644)  NO, IS INA
1143 1634          260 C=N                    GET ALPHA STRING FROM N
1144 1635         1434 PT=     1              POINT AT LOWEST BYTE
1145 1636         1352 ? C#0   WPT            7 CHARACTERS YET ?
1146 1637         1367 GOC     INAECO (1575)  YES, IGNORE THE REST
1147 1640          252 AC EX   WPT            C[1:0] = ALPHA CHARACTER
1148 1641         1074 RCR     2              ROTATE STRING 1 BYTE RIGHT
1149 1642          160 N=C                    SAVE APPENDED STRING TO N
1150 1643         1323 GOTO    INAECO (1575)  ECHO AND READ NEXT FRAME
1151 1644 INAD25     1 GOSUB   APPEND         APPEND IT TO ALPHA REG
1151 1645            0                        *MAINFRAME: CN11, @0416
1152 1646          332 C=B     M              GET CHARACTER COUNT FROM B
1153 1647         1172 C=C-1   M              SEE IF ALPHA IS FULL ?
1154 1650         1547 GOC     INANRD (1624)  ALREADY READ 24 CHARACTERS
1155 1651          372 BC EX   M              SAVE CHARACTER COUNT TO B
1156 1652 INAD30 1233 GOTO    INAECO (1575)  ECHO AND READ NEXT FRAME
1157 1653 IND10    460 LDI                    FIND A DIGIT
1158 1654           60 CON     48             @60 = 48 = ASCII 0 (ZERO)
1159 1655         1406 ? A<C   X              IS IT A NUMBER ?
1160 1656          117 GOC     ASCDG2 (1667)  SEE IF MINUS SIGN OR D.P.
1161 1657          460 LDI                    LOAD LOW 12 BITS OF C WITH
1162 1660           72 CON     58             ONE MORE THAN ASCII 9
1163 1661         1406 ? A<C   X              IS IT A NUMBER ?
1164 1662          223 GONC    ASCDG4 (1704)  SEE IF EXPONENT PRESENT
1165 1663          460 LDI                    YES, IT IS A NUMBER
1166 1664           40 CON     @40            CONVERT TO CN CODE
1167 1665         1106 C=A-C   X              CHARACTER NUMBER CODE
1168 1666          243 GONC    CHRCON (1712)  INSERT A # AND GET NXTCHR
```

```
  1169 1667 ASCDG2  460 LDI                  LOAD LOW 12 BITS OF C WITH
  1170 1670          55 CON    45            45 = ASCII (-) MINUS SIGN
  1171 1671        1546 ? A#C  X             IS IT A MINUS SIGN ?
  1172 1672          47 GOC    ASCDG3 (1676) NO, NOT A MINUS SIGN
  1173 1673         460 LDI                  YES, MINUS SIGN FOUND
  1174 1674          34 CON    @34           CN KEY CODE FOR MINUS SIGN
  1175 1675         153 GOTO   CHRCON (1712) XEQ CHS AND GET NEXT CHAR
  1176 1676 ASCDG3 1046 C=C+1  X             C[X] = 46 = ASCII PERIOD
  1177 1677        1546 ? A#C  X             IS IT A PERIOD ?
  1178 1700        1527 GOC    INAD30 (1652) NO, NOT A PERIOD
  1179 1701         460 LDI                  CONVERT TO CN CODE
  1180 1702          32 CON    @32           CN KEY CODE FOR PERIOD
  1181 1703          73 GOTO   CHRCON (1712) INSERT "." AND GET NXTCHR
  1182 1704 ASCDG4  460 LDI                  LOAD LOW 12 BITS OF C WITH
  1183 1705         105 CON    69            69 = ASCII E (FOR EEX)
  1184 1706        1546 ? A#C  X             IS IT EEX ?
  1185 1707        1437 GOC    INAD30 (1652) NO, INVALID CHARACTER
  1186 1710         460 LDI                  LOAD LOW 12 BITS OF C WITH
  1187 1711          33 CON    @33           CN KEY CODE FOR EEX
  1188 1712 CHRCON  130 G=C                  PLACE DIGIT INTO G AND PRAY
  1189 1713           1 GOSUB  DIGENT        DIGIT ENTRY SUBROUTINE
  1189 1714           0                      *MAINFRAME:  CN2, @0067
  1190 1715 INVCHR 1353 GOTO   INAD30 (1652) ECHO AND READ NEXT FRAME
*
  1192                  ENTRY  INADEX
  1193 1716 INADEX  406 A=C    X             MOVE C[X] INTO A[X]
  1194 1717         460 LDI                  LOAD LOW 12 BITS OF C WITH
  1195 1720         100 CON    @100          END TRANSMISSION OK (ETO)
  1196 1721        1546 ? A#C  X             IS IT AN ETO ?
  1197 1722          27 GOC    INERCK (1724) NO, ERROR
  1198 1723        1104 S9=    0             CLEAR ERROR FLAG S9
  1199 1724 INERCK  414 ?S8=1                INPUT DECIMAL (IND) ?
  1200 1725          47 GOC    INEK10 (1731) YES, GOTO UNTALK & ERRCHK
  1201 1726          14 ?S3=1                FIND ID ?
  1202 1727           1 GOLC   UNT           YES, SEND UNTALK & EXIT
  1202 1730           3                      *ILCAS&CTL:  CS0, @0254
  1203 1731 INEK10    1 GOSUB  UNTCHK        UNTALK & ERROR CHECKING
  1203 1732           0                      *ILCAS&CTL:  CS1, @1745
  1204 1733         414 ?S8=1                INPUT DECIMAL (IND) ?
  1205 1734        1640 RTN NC               NO, INPUT ALPHA, RETURN
  1206 1735         614 ?S11=1               PUSH FLAG SET ?
  1207 1736           1 GSUBC  R^SUB         PUSH STACK IF YES
  1207 1737           1                      *MAINFRAME:  CN5, @0355
  1208 1740           1 GOLONG NOREG9        NORMALIZE REG 9, SEND TO X
  1208 1741           2                      *MAINFRAME:  CN2, @0536
  1209                  ENTRY  R5-R6         COPY R5R/W TO R6R/W
*
  1211 1742 R5-R6   544 C=HPIL 5             READ HPIL SCRATCH R5R/W
  1211 1743         572                      (INSERTED BY ASSEMBLER)
  1211 1744         503                      (INSERTED BY ASSEMBLER)
  1212 1745        1600 HPIL=C 6             WRITE HPIL SCRATCH R6R/W
  1213 1746        1740 RTN                  END OF R5 TO R6 ROUTINE
*
  1215                  FILLTO @1746
*****************************************************************
* CX<128 - CONVERT X TO BINARY, CHECK X < 128                   *
*                                                               *
* USES:    A[X], C, NO STS, NO PT, 2 ADDITIONAL SUBROUTINE LEVELS *
*                                                               *
* INPUT:   CONTENTS OF X-REG, CHIP 0 ENABLED, HEX MODE          *
```

```
*                                                                     *
* OUTPUT:  A[X] = BINARY NUMBER < 128, CHIP 0 ENABLED, HEX MODE       *
*                                                                     *
***********************************************************************
  1226                    ENTRY  CX<128
  1227 1747 CX<128     1 GOSUB  CONV3D         CONVERT X-REG TO BINARY
  1227 1750             0                      *ILCAS&CTL:  CS3, @1465
  1228 1751           406 A=C    X              A[X] = BINARY NUMBER
  1229 1752           460 LDI                   LOAD LOW 12 BITS OF C WITH
  1230 1753           200 CON    128            MAXIMUM ALLOWABLE VAL + 1
  1231 1754          1406 ? A<C  X              NUMBER < 128 ?
  1232 1755          1540 RTN C                 YES, RETURN WITHOUT ERROR
  1233 1756             1 GOLONG ERRDE          NO, SHOW MESSAGE: DATA ERR
  1233 1757             2                       *MAINFRAME: CN10, @0055
*
***********************************************************************
* ACKX  -  ALPHA CHECK OF X-REGISTER (ERRORS IF ALPHA)                *
* ACKC  -  ALPHA CHECK OF C-REGISTER (ERRORS IF ALPHA)                *
*                                                                     *
* USES:    C, NO PT, NO STATUS, NO ADDITIONAL SUBROUTINE LEVELS       *
*                                                                     *
* INPUT:   CHIP 0 ENABLED, HEX MODE                                   *
*                                                                     *
* OUTPUT:  C = X|C REGISTER, EXCEPT THE SIGN FIELD HAS BEEN DESTROYED  *
***********************************************************************
  1245                    ENTRY  ACKX
  1246                    ENTRY  ACKC
  1247 1760 ACKX       370 C=REGN 3               GET CONTENTS OF X-REGISTER
  1248 1761 ACKC      1176 C=C-1  S               ALPHA HAS SIGN DIGIT OF 1
  1249 1762           1176 C=C-1  S               CHECK FOR ALPHA DATA
  1250 1763           1640 RTN NC                 NOT ALPHA, RETURN NO ERROR
  1251 1764              1 GOSUB  IFC             SEND INTERFACE CLEAR CMD
  1251 1765              0                        *ILCAS&CTL:  CS3, @0113
  1252 1766              1 GOLONG ERRAD           ERROR = ALPHA DATA
  1252 1767              2                        *MAINFRAME:  CN5, @0342
*
*
  1255                    ENTRY  RENTPH           PATCH OF "RDENT"
*
  1257 1770 RENTPH       1 GOSUB  CSSTAS          READ CASSETTE STATUS
  1257 1771              0                        *ILCAS&CTL:  CS0, @0700
  1258 1772            214 ?S5=1                  CASSETTE BUSY ?
  1259 1773           1757 GOC    RENTPH (1770)   YES, WAIT UNTIL NOT BUSY
  1260 1774              1 GOSUB  CSSTCK          CHECK FOR A READ ERROR
  1260 1775              0                        *ILCAS&CTL:  CS1, @0040
  1261 1776              1 GOLONG RDENT           READ CASSETTE FILE ENTRY
  1261 1777              2                        *ILCAS&CTL:  CS2, @0241
*
*
  1264                    UNLIST
  1267                    END

  ERRORS :      0
```

```
SYMBOL TABLE (SCPL1B - ILCAS&CTL QUAD 0 = CS0 = ADDRESSES @70000-71777)

AAU10    212   -   217
AAU20    220   -   213
AAU25    222   -   235  216  176
ACKC    1761   -
ACKX    1760   -
AOUT1   1064   -
AOUT10  1115   -  1172
AOUT12  1137   -  1134
AOUT14  1142   -
AOUT15  1143   -
AOUT16  1153   -  1147
AOUT17  1160   -  1152
AOUT18  1162   -  1127
AOUT20  1164   -  1161 1140 1124 1101
AOUT30  1203   -  1175
AOUT31  1222   -  1211 1177
AOUT32  1224   -  1234 1204 1136
AOUT33  1232   -  1241
AOUT34  1233   -
AOUT35  1235   -  1230
AOUT40  1242   -  1225
AOUT50  1246   -  1276
AOUT55  1254   -  1243
AOUTFL  1066   -
AOUTIN  1266   -
ASCDG2  1667   -  1656
ASCDG3  1676   -  1672
ASCDG4  1704   -  1662
ASP      205   -
ASP00    224   -   221
ASP10    231   -   236
ASP30    237   -   232
ASP40    241   -
ASP50    246   -   240
ASP60    247   -   245
AUTOIO  1373   -
C=R2     437   -   507  445  435
CHKLAD  1310   -
CHRCON  1712   -  1703 1675 1666
CKCRLF  1614   -  1611
CSSTAS   700   -
CSTNOP   142   -
CX<128  1747   -
DDT0     320   -
DSPERJ  1264   -  1334
DTFLOW   332   -
FDEV10   607   -   630
FDEV20   621   -   640
FDEV36   641   -   667  635
FDEV38   644   -   633
FDEV40   652   -   670  647  620
FDEV50   661   -   627
FDEV60   664   -   655
FLAG32  1374   -  1364
FLNMER  1256   -  1201
FNDCAS   566   -
```

```
FNDDEV   601   -   574
FNDPTR   575   -
FNSTS    702   -
FNSTSA   703   -   774  701
FRTR35   631   -   616
FS100   1061   -  1054
FSP04    733   -   731
FSP06    737   -   734
FSP10    746   -   720
FSP30    755   -   761
FSP45    762   -   756
FSP48    766   -  1032
FSP50    767   -   772
FSP60   1017   -   770
FSP65   1033   -  1026
FSP68   1036   -  1024
FSP70   1040   -  1044
FSP75   1045   -  1041
FSP80   1051   -  1013
FSP85   1053   -   776
FSPER    773   -  1052 1043 1015  760
FSPEX    775   -  1050
FSPEX4   777   -   745
FSPEX6  1012   -  1016
GETDEV   520   -
GTD#20   534   -   531
IDYTST   175   -   200
INA     1545   -
INAD    1553   -  1541
INAD00  1571   -  1567
INAD2   1563   -
INAD20  1630   -  1623 1613
INAD22  1632   -
INAD25  1644   -  1633
INAD30  1652   -  1715 1707 1700
INADEX  1716   -
INADRD  1577   -  1574
INAECO  1575   -  1652 1643 1637 1617
INANRD  1624   -  1650
IND10   1653   -  1631
INDX    1525   -
INEK10  1731   -  1725
INERCK  1724   -  1722 1627
INVCHR  1715   -
LAD      342   -
LADER   1322   -  1316 1311
LADERM  1327   -  1323
LISTEN   335   -
LISTER  1506   -  1511
LISTNR  1500   -
LOCAL   1436   -
MANIO   1363   -
NATN10   357   -   363
NATNRD   352   -
NRD      364   -
NRDC     366   -
NXTD25   550   -
NXTD30   555   -   552
NXTDEI   545   -
NXTDEV   541   -
```

```
OOP10   1060   -  1056
OOPCHK  1055   -
OUTA    1106   -
PILEN    143   -
PILPRM   140   -
R5-R6   1742   -
RDDFRM   420   -
RDFM10   421   -   426  424
RDFM20   430   -   422
RDFMER   436   -   500  463  460  427  415  362
RDTYPC   466   -
RDTYPE   464   -
REMOTE  1447   -
RENTPH  1770   -  1773
RSIDY   1410   -
RTNP+2   656   -   565
RTPACH   671   -
RTPHRT   512   -
RTYP10   473   -   517  513
RTYP20   474   -   501
RTYP30   502   -   475
SCHDEV  1335   -
SCMD     272   -   345  334  325  322  261  256
SCMD15   275   -
SCMD20   276   -   305
SCMD26   312   -   316
SCMD30   315   -   277
SCMDER   306   -   317  223
SDAT10   452   -   457  455
SDAT20   461   -   453
SDATA    450   -
SDATA0   446   -
SEKSUB   326   -
SELD10  1320   -  1326
SELECT  1305   -
SETIDY  1413   -  1355
SF51    1345   -
SF5521  1343   -
SFRMC    274   -
SFRTN   1354   -
SNCMD   1456   -  1440 1430
SNDATA   346   -
SRWRT    323   -
TAD      265   -
TALKER   262   -
TRIGER  1426   -
UNL      257   -
UNLCHK  1466   -
UNLRSF   443   -
UNT      254   -
WRTDAT   330   -
```

ENTRY TABLE (SCPL1B - ILCAS&CTL QUAD 0 = CS0 = ADDRESSES @70000-71777)

ACKC	1761	-
ACKX	1760	-
AOUT1	1064	-
AOUTFL	1066	-
AOUTIN	1266	-
ASP	205	-
ASP10	231	-
AUTOIO	1373	-
C=R2	437	-
CHKLAD	1310	-
CSSTAS	700	-
CSTNOP	142	-
CX<128	1747	-
DDT0	320	-
DTFLOW	332	-
FDEV20	621	-
FLNMER	1256	-
FNDCAS	566	-
FNDPTR	575	-
FNSTS	702	-
FNSTSA	703	-
FS100	1061	-
GETDEV	520	-
INA	1545	-
INAD2	1563	-
INADEX	1716	-
INADRD	1577	-
INDX	1525	-
LAD	342	-
LADERM	1327	-
LISTEN	335	-
LISTNR	1500	-
LOCAL	1436	-
MANIO	1363	-
NATNRD	352	-
NRD	364	-
NRDC	366	-
NXTDEI	545	-
NXTDEV	541	-
OOPCHK	1055	-
OUTA	1106	-
PILEN	143	-
PILPRM	140	-
R5-R6	1742	-
RDDFRM	420	-
RDFMER	436	-
RDTYPC	466	-
RDTYPE	464	-
REMOTE	1447	-
RENTPH	1770	-
RSIDY	1410	-
RTNP+2	656	-
RTPACH	671	-
RTPHRT	512	-
SCHDEV	1335	-
SCMD	272	-

```
SCMD15   275   -
SCMD20   276   -
SCMD30   315   -
SCMDER   306   -
SDATA    450   -
SDATA0   446   -
SEKSUB   326   -
SELECT  1305   -
SF51    1345   -
SF5521  1343   -
SFRMC    274   -
SNDATA   346   -
SRWRT    323   -
TAD      265   -
TALKER   262   -
TRIGER  1426   -
UNL      257   -
UNLCHK  1466   -
UNLRSF   443   -
UNT      254   -
WRTDAT   330   -
```

EXTERNAL REFERENCES (SCPL1B - ILCAS&CTL QUAD 0 = CS0 = ADR @70000-71777)

```
AOUTIN  1064  1113
AOUTIN  1065  1114
APPEND  1644
APPEND  1645
AUTOIO    71
AUTOIO    72
CASSET     3
CASSET     2
CLA     1557
CLA     1560
CONV3D  1305  1500  1747
CONV3D  1306  1501  1750
CREATF     5
CREATF     4
CSSTAS  1770
CSSTAS  1771
CSSTCK  1774
CSSTCK  1775
CSTNOP    65
CSTNOP    64
DIGENT  1713
DIGENT  1714
DIR        7
DIR        6
DSPERR  1264
DSPERR  1265
ERRAD   1766
ERRAD   1767
ERRDE   1756
ERRDE   1757
FINDID    73
FINDID    72
FNDPTR  1410
FNDPTR  1411
FNSTSA   652
FNSTSA   653
GETDEV   605  1341
GETDEV   606  1342
GTBYTA  1117
GTBYTA  1120
IFC     1764
IFC     1765
INA       75
INA       74
INADEX  1602
INADEX  1603
INCADA  1115
INCADA  1116
INDX      77
INDX      76
INSTAT   101
INSTAT   100
LADERM  1506
LADERM  1507
LDSST0   644  1343
LDSST0   645  1344
```

```
LISTEN  1110  1461  1520
LISTEN  1111  1462  1521
LISTNR   103
LISTNR   102
LOCAL    105
LOCAL    104
MANIO    107
MANIO    106
NATNRD  1571
NATNRD  1572
NEWTAP    11
NEWTAP    10
NOREG9  1740
NOREG9  1741
NRDC    1625
NRDC    1626
NXTDEV   625
NXTDEV   626
OUTA     111
OUTA     110
PILEN    566   577  1335  1514
PILEN    567   600  1336  1515
PILPRM    67
PILPRM    66
PLERCK   572  1254  1337  1470  1516
PLERCK   573  1255  1340  1471  1517
PLEREX  1256  1327
PLEREX  1257  1330
PURGEF    13
PURGEF    12
PWRDN    113
PWRDN    112
PWRUP    115
PWRUP    114
RDDFRM   364   405  1577
RDDFRM   365   406  1600
RDENT   1776
RDENT   1777
RDTYPE   611
RDTYPE   612
READA     15
READA     14
READK     17
READK     16
READP     21
READP     20
READR     23
READR     22
READRX    25
READRX    24
READS     27
READS     26
READSB    31
READSB    30
REMOTE   117
REMOTE   116
RENAME    33
RENAME    32
RTPACH   510
RTPACH   511
```

```
WRTS      61
WRTS      60
ZERO      63
ZERO      62

End of VASM assembly
********************************************************************************
********************************************************************************
********************************************************************************

VASM ROM ASSEMBLY          REV.  6/81A            HP-82160A HP-IL MODULE

OPTIONS: L C S          HP-IL CAS&CTL             ADDRESSES @72000-73777

     2                     FILE   SCPL2B             ILCAS&CTL QUAD 1 = CS1
     3    0          210 CON    @210          H
     4    1           61 CON    @61           1
     5    2           40 CON    @40                BLANK
     6    3           24 CON    @24           T
     7    4           23 CON    @23           S
     8    5           40 CON    @40                BLANK
     9    6           23 CON    @23           S
    10    7           23 CON    @23           S
    11   10            1 CON    @01           A
    12   11           15 CON    @15           M
    13   12           55 CON    @55           -
    14
    15       CASSET      ENTRY  CASSET
    16
*
    18   13 NOTAPE     1 GOSUB  PLEREX        ERROR CODE 4-7: NO MEDM
    18   14            0                      *ILCAS&CTL:  CS1, @1663
    19   15           16 CON    @16           N
    20   16           17 CON    @17           O
    21   17           40 CON    @40                BLANK
    22   20           15 CON    @15           M
    23   21            5 CON    @05           E
    24   22            4 CON    @04           D
    25   23         1015 CON    @1015         M
    26   24            1 GOLONG CSEREX        CASSETTE ERROR EXIT
    26   24            2                      *ILCAS&CTL:  CS3, @0376
********************************************************************************
* PARWRT - SEND COMMAND - PARTIAL WRITE                                        *
* ASSUMES: CASSETTE IS A LISTENER ALREADY                                      *
* RETURNS: CASSETTE AS A TALKER                                                *
*                                                                              *
* USES:    A, C, +1 SUBROUTINE LEVEL                                           *
********************************************************************************
    34                     ENTRY  PARWRT
    35                     ENTRY  SCMDWT
*
    37   26 PARWRT   460 LDI                   DEV DEPENDENT LISTENER 6
    38   27          246 CON    @246           DDL 06 (PARTIAL WRITE)
    39   30 SCMDWT     1 GOSUB  SCMD           SEND OUT A COMMAND
    39   31            0                      *ILCAS&CTL:  CS0, @0272
********************************************************************************
* THE FOLLOWING ROUTINE WILL READ CASSETTE STATUS AND WAIT UNTIL               *
* CASSETTE IS DONE WITH LAST OPERATION                                         *
*                                                                              *
* WAITS  - READ CASSETTE STATUS UNTIL IT IS DONE WITH LAST OPERATION           *
```

```
*          AND CHECK ERROR                                              *
*                                                                       *
* USES:    A, C, S[7:0], +1 SUBROUTINE LEVEL                            *
*************************************************************************
    49                   ENTRY  WAITS
    50                   ENTRY  CSERR
    51                   ENTRY  CSSTCK
    52                   ENTRY  TAPERR
*
    54   32 WAITS     1 GOSUB  CSSTAS        READ CASSETTE STATUS
    54   33           0                      *ILCAS&CTL:  CS0, @0700
    55   34           1 GOSUB  PLERCK        PRINTER ERROR CHECK
    55   35           0                      *ILCAS&CTL:  CS1, @1747
    56   36         214 ?S5=1                STILL BUSY ?
    57   37        1737 GOC    WAITS  (  32) YES, LET'S WAIT
    58   40 CSSTCK  114 ?S4=1                ANY CASSETTE ERROR ?
    59   41        1640 RTN NC               NO ERROR, RETURN
*************************************************************************
* WHEN THERE IS AN ERROR, THE LOWER 4 BITS OF THE CASSETTE STATUS       *
* IS USED TO PRESENT AN ERROR NUMBER AS FOLLOWS:                        *
*   1 (0001) - EOT   ..............  DRIVE ERR                          *
*   2 (0010) - STALL ..............  DRIVE ERR                          *
*   4 (0100) - DOOR OPEN ..........  NO MEDM                           *
*   5 (0101) - NO TAPE ............  NO MEDM                           *
*   7 (0111) - NEW TAPE ...........  NO MEDM                           *
*   8 (1000) - TIME OUT ...........  MEDM ERR                          *
*   9 (1001) - RECORD # ERROR .....  MEDM ERR                          *
*   A (1010) - CHECKSUM ERROR .....  MEDM ERR                          *
*************************************************************************
    72   42 CSERR    14 ?S3=1                TAPE ERROR ?
    73   43         137 GOC    TAPERR (  56) YES, HANDLE IT
    74   44        1014 ?S2=1                NO TAPE ?
    75   45        1467 GOC    NOTAPE (  13) YES, HANDLE IT
    76   46           1 GOSUB  PLEREX        ERROR CODE 0-3: DRIVE ERR
    76   47           0                      *ILCAS&CTL:  CS1, @1663
    77   50           4 CON    @04           D
    78   51          22 CON    @22           R
    79   52          11 CON    @11           I
    80   53          26 CON    @26           V
    81   54        1005 CON    @1005         E
    82   55          73 GOTO   DSERJ  (  64) DISPLAY MESSAGE: ERR
    83   56 TAPERR    1 GOSUB  PLEREX        ERROR CODE 8-B: MEDM ERR
    83   57           0                      *ILCAS&CTL:  CS1, @1663
    84   60          15 CON    @15           M
    85   61           5 CON    @05           E
    86   62           4 CON    @04           D
    87   63        1015 CON    @1015         M
    88   64 DSERJ     1 GOLONG DSPERR        DISPLAY MESSAGE: ERR
    88   65           2                      *ILCAS&CTL:  CS3, @0405
*
*************************************************************************
* DIR - CASSETTE DIRECTORY FUNCTION                                     *
*************************************************************************
*
    94                   ENTRY  DIR
    95                   ENTRY  DIRROM
    96                   ENTRY  DIR150
*
    98   66         222 CON    @222          R
    99   67          11 CON    @11           I
```

```
100   70           4 CON    @04           D
101   71 DIR       1 GOSUB  CHKCST        SEE IF CASSETTE IS READY
101   72           0                      *ILCAS&CTL:  CS3, @0335
102   73           1 GOSUB  CLA           CLEAR ALPHA REGISTER
102   74           0                      *MAINFRAME:  CN4, @0321
103   75           1 GOSUB  BLDAPH        PRINT DIRECTORY HEADER
103   76           0                      *ILCAS&CTL:  CS1, @0273
104   77         116 CON2   4      14      N
105  100         101 CON2   4      1       A
106  101         115 CON2   4      13      M
107  102         105 CON2   4      5       E
108  103          40 CON2   2      0            BLANK
109  104          40 CON2   2      0            BLANK
110  105          40 CON2   2      0            BLANK
111  106          40 CON2   2      0            BLANK
112  107         124 CON2   5      4       T
113  110         131 CON2   5      9       Y
114  111         120 CON2   5      0       P
115  112         105 CON2   4      5       E
116  113          40 CON2   2      0            BLANK
117  114          40 CON2   2      0            BLANK
118  115          40 CON2   2      0            BLANK
119  116          40 CON2   2      0            BLANK
120  117          40 CON2   2      0            BLANK
121  120         122 CON2   5      2       R
122  121         105 CON2   4      5       E
123  122         107 CON2   4      7       G
124  123         523 CON    @523          S
125  124           1 GOSUB  PRT11?        SEE IF WE NEED TO PRINT
125  125           0                      *ILCAS&CTL:  CS1, @0424
126  126 DIRROM    1 GOSUB  FNDCAS        LOOK FOR CASSETTE AGAIN
126  127           0                      *ILCAS&CTL:  CS0, @0566
127  130        1123 GOTO   CSERR  (  42) P+1 - NO CASSETTE FOUND
128  131 DIR20     1 GOSUB  SEEKR2        P+2 - SEEK & READ REC.2
128  132           0                      *ILCAS&CTL:  CS3, @1557
129  133 DIR150    1 GOSUB  RENTPH        READ ONE FILE ENTRY
129  134           0                      *ILCAS&CTL:  CS0, @1770
130  135         630 C=M                  GET FILE NAME
131  136        1056 C=C+1                REACH END OF DIRECTORY ?
132  137           1 GOLC   UNT           YES, ALL DONE, SEND UNTALK
132  140           3                      *ILCAS&CTL:  CS0, @0254
133  141         260 C=N                  GET FILE INFO FROM N-REG
134  142        1376 ? C#0  S             IS IT A PURGED FILE ?
135  143        1703 GONC   DIR150 ( 133) YES, READ NEXT ENTRY
136  144           1 GOSUB  CLA           NO, CLEAR ALPHA REGISTER
136  145           0                      *MAINFRAME:  CN4, @0321
137  146         630 C=M                  GET FILE NAME AGAIN
138  147         550 REGN=C 5             PUT NAME TO ALPHA REG.
139  150           1 GOSUB  BLDAPH        PAD IN ONE BLANK
139  151           0                      *ILCAS&CTL:  CS1, @0273
140  152         440 CON    @440          TERMINATING SPACE CHAR
*
142  153 FILTYP  260 C=N                  GET FILE INFO FROM N-REG
143  154         106 C=0    X             CLEAR FILE SIZE FROM C[X]
144  155        1374 RCR    13            C[X] = FILE TYPE
145  156         406 A=C    X             SAVE FILE TYPE TO A[X]
146  157           1 GOSUB  TYPASC        LOAD THE ASCII FILE TYPE
146  160           0                      *ILCAS&CTL:  CS1, @0260
147  161          10 CON    8             PROGRAM FILE TYPE
148  162         120 CON    @120          P
```

```
   149  163          522 CON    @522           R
   150  164          303 GOTO   FLTYOP ( 214) CHECK FILE PROTECTION
   151  165           15 CON    13             DATA FILE TYPE
   152  166          104 CON    @104           D
   153  167          501 CON    @501           A
   154  170          243 GOTO   FLTYOP ( 214) CHECK FILE PROTECTION
   155  171            4 CON    4              WRITE ALL FILE TYPE
   156  172          127 CON    @127           W
   157  173          501 CON    @501           A
   158  174          203 GOTO   FLTYOP ( 214) CHECK FILE PROTECTION
   159  175            5 CON    5              KEY FILE TYPE
   160  176          113 CON    @113           K
   161  177          505 CON    @505           E
   162  200          143 GOTO   FLTYOP ( 214) CHECK FILE PROTECTION
   163  201            6 CON    6              STATUS FILE TYPE
   164  202          123 CON    @123           S
   165  203          524 CON    @524           T
   166  204          103 GOTO   FLTYOP ( 214) CHECK FILE PROTECTION
   167  205            1 CON    1              ASCII DATA FILE
   168  206          101 CON    @101           A
   169  207          523 CON    @523           S
   170  210           43 GOTO   FLTYOP ( 214) CHECK FILE PROTECTION
   171  211            0 CON    0              UNRECOGNIZED FILE
   172  212           77 CON    @77            ?
   173  213          477 CON    @477           ?
   174  214 FLTYOP   260 C=N                   GET FILE PROTECTION INFO
   175  215         1574 RCR    12             C[1:0] = FILE TYPE/PROTECT
   176  216         1530 ST=C                  MOVE C[1:0] TO STATUS AREA
* S3 = USER FILE SECURE
* S1 = AUTO RUN
* S0 = PRIVATE
   180  217           14 ?S3=1                 FILE SECURED ?
   181  220           53 GONC   FLTP20 ( 225) NO, CHECK A AND/OR P FLAGS
   182  221 FLTPS     1 GOSUB  BLDAPH         ADD ,S TO FILE TYPE STRING
   182  222            0                      *ILCAS&CTL:  CS1, @0273
   183  223           54 CON    @54            COMMA
   184  224          523 CON    @523           S
   185  225 FLTP20 1414 ?S1=1                  IS FILE AUTO RUN ?
   186  226           53 GONC   FLTP30 ( 233) NO, CHECK FOR P FLAG
   187  227            1 GOSUB  BLDAPH         ADD ,A TO FILE TYPE STRING
   187  230            0                      *ILCAS&CTL:  CS1, @0273
   188  231           54 CON    @54            COMMA
   189  232          501 CON    @501           A
   190  233 FLTP30 1614 ?S0=1                  IS FILE PRIVATE ?
   191  234           53 GONC   FLTP40 ( 241) NO, DISP FILE NAME & TYPE
   192  235            1 GOSUB  BLDAPH         ADD ,P TO FILE TYPE STRING
   192  236            0                      *ILCAS&CTL:  CS1, @0273
   193  237           54 CON    @54            COMMA
   194  240          520 CON    @520           P
   195  241 FLTP40   404 S8=    0              S8=0 : SCROLL, NO PROMPT
   196  242         1110 S9=    1              S9=1 : KEYBOARD WAS RESET
   197  243            1 GOSUB  ARGOUT         DISPLAY FILE NAME & TYPE
   197  244            0                      *MAINFRAME: CN11, @0020
   198  245            1 GOSUB  DSDLY          DELAY .6 SECONDS
   198  246            0                      *ILCAS&CTL:  CS1, @1466
   199  247 PADBLK   770 C=REGN 7              C[1:0] = SCRATCH R7R/W
   200  250         1074 RCR    2              C[13:12] = PREV. C[1:0]
   201  251         1434 PT=    1              POINT AT LOWEST BYTE OF C
   202  252         1352 ? C#0  WPT            16 CHARS IN YET ?
   203  253          347 GOC    PRBIN# ( 307) YES, PRINT OUT RESULTS
```

```
   204  254           1 GOSUB  BLDAPH        NO, PAD WITH A BLANK
   204  255           0                      *ILCAS&CTL:  CS1, @0273
   205  256         440 CON    @440           TERMINATING SPACE CHAR
   206  257        1703 GOTO   PADBLK ( 247) CONTINUE PADDING BLANKS
   207                ENTRY  TYPASC
   208  260 TYPASC  660 C=STK                 POP RTN ADDR TO C[6:3]
   209  261 TYPS10 1460 CXISA                 C[2:0] = VALUE AT C[6:3]
   210  262        1072 C=C+1  M              POINT TO ASCII CHARACTERS
   211  263        1346 ? C#0  X              END OF TABLE ?
   212  264         103 GONC   BLDAPC ( 274) YES, UNRECOGNIZED TYPE
   213  265        1546 ? A#C  X              FOUND THE TYPE ?
   214  266          63 GONC   BLDAPC ( 274) YES, FILE TYPE IS A MATCH
   215  267        1072 C=C+1  M              SKIP OVER 3 BYTES
   216  270        1072 C=C+1  M
   217  271        1072 C=C+1  M
   218                LEGAL                  (CLEAR THE CARRY FLAG)
   219  272        1673 GOTO   TYPS10 ( 261) KEEP LOOKING FOR FILE TYPE
*
*************************************************************************
* BLDAPH - BUILD ASCII CHARACTER INTO ALPHA REGISTER                     *
*                                                                        *
* CALLING SEQUENCE:                                                      *
*         GOSUB  BLDAPH                                                  *
*         CON    (ONE ASCII CODE)                                        *
*         CON    .                                                       *
*          .     .                                                       *
*          .     .                                                       *
*         CON    00      (END OF ASCII TABLE)                            *
*                                                                        *
* BLDAPC - SAME AS BLDAPH EXCEPT RETURN ADDR(STK) ALREADY IN C-REG       *
*          AS THE FIRST ASCII CHARACTER                                  *
* USES:    A, C, G, PT, +1 SUBROUTINE LEVEL                              *
*************************************************************************
   236                ENTRY  BLDAPH
   237                ENTRY  BLDAPC
*
   239  273 BLDAPH  660 C=STK                 POP RTN ADDRESS TO C[6:3]
   240  274 BLDAPC 1460 CXISA                 C[2:0] = VALUE AT C[6:3]
   241  275        1072 C=C+1  M              POINT TO ASCII CHARACTERS
   242  276         560 STK=C                 PUSH NEXT ADDRESS ON STACK
   243  277        1634 PT=    0              POINT AT LOWEST BYTE OF C
   244  300         130 G=C                   SAVE LOWEST BYTE IN REG G
   245  301        1366 ? C#0  XS             LAST CHARACTER ?
   246  302           1 GOLC   APPEND         YES, APPEND CHAR AND EXIT
   246  303           3                       *MAINFRAME: CN11, @0416
   247  304           1 GOSUB  APPEND         NO, APPEND CHAR & RETURN
   247  305           0                       *MAINFRAME: CN11, @0416
   248  306        1653 GOTO   BLDAPH ( 273) SEND NEXT CHARACTER
* PRBIN# - CONVERT TYPE 8 PROGRAM FILE BYTE COUNT TO REGISTER COUNT
* INPUT:   N HOLDS FILE INFORMATION, OUTPUT: A[2:0] = REGISTER COUNT
   251  307 PRBIN# 1334 PT=    13             POINT AT C[S] DIGIT
   252  310        1020 LC     8              8 IS PROGRAM FILE TYPE
   253  311         436 A=C    S              SAVE FILE TYPE 8 IN A[S]
   254  312         260 C=N                   C[3:0] = FILE SIZE
   255  313          34 PT=    3              POINT AT FILE SIZE AREA
   256  314         412 A=C    WPT            A[3:0] = # OF PROG BYTES
   257  315        1576 ? A#C  S              IS THIS A PROGRAM FILE ?
   258  316         127 GOC    B-DEC  ( 330) NO, ANOTHER TYPE, EXIT
   259  317         116 C=0                   YES, CLEAR ACCUMULATOR
   260  320         460 LDI                   CONVERT BYTES TO REGISTERS
```

```
   261  321           7 CON    7              7 BYTES EQUALS 1 REGISTER
   262  322         646 A=A-1  X              SUBTRACT 1 FROM FILE SIZE
   263  323 B-RG10 1072 C=C+1  M              ADD 1 TO NUMBER OF REGS
   264  324         706 A=A-C  X              SUBTRACT 7 FROM BYTE COUNT
   265  325        1763 GONC   B-RG10 ( 323) KEEP COUNTING UNTIL DONE
   266  326          74 RCR    3              C[2:0] = REGISTER COUNT
   267  327         406 A=C    X              A[2:0] = REGISTER COUNT
*******************************************************************
* B-DEC  - CONVERT A BINARY VALUE TO A 5-DIGIT DECIMAL NUMBER     *
* INPUT:   A[3:0]  = BINARY VALUE                                 *
* OUTPUT:  A[12:8] = DECIMAL DIGITS                               *
* USES:    A, B, C, PT, +1 SUB LEVEL                              *
*******************************************************************
   274  330 B-DEC   216 B=A                B[2:0] = BINARY VALUE
   275  331         256 AC EX              SAVE C INTO REGISTER A
   276  332           1 GOSUB  BINBD0      USE "GENNUM" TO DO BIN-DEC
   276  333           0                    *ILCAS&CTL:  CS3, @1724
* "GENNUM" WILL ONLY CONVERT THE BINARY IN A[X] TO DECIMAL, BUT OUR
* DATA FILE MAY BE LARGER THAN 4096 REGISTERS (HEX 1000), SO WE HAVE
* TO BE ABLE TO HANDLE THIS CASE.
   280  334         106 C=0    X              CLEAR ACCUMULATOR EXPONENT
   281  335        1160 DADD=C                ENABLE CHIP 0 AGAIN
   282  336        1334 PT=    13             POINT AT MANTISSA SIGN
   283  337         420 LC     4              NUMBER OF DIGITS IS 4
   284  340         436 A=C    S              SAVE # OF DIGITS TO A[S]
   285  341         336 C=B    S              C[S]= # OF OUTPUT DIGITS
   286  342        1136 C=A-C  S              C[S]= 4 - # OF OUTPUT DGTS
   287  343 B-D10  1632 A SR   M              SHIFT IN LEADING ZEROS TO
   288  344        1176 C=C-1  S               MAKE NUMBER A 5-DIGIT #
   289  345        1763 GONC   B-D10  ( 343) SHIFT IN NEXT LEADING 0
   290  346          20 LC     0
   291  347         420 LC     4
   292  350          20 LC     0
   293  351        1120 LC     9
   294  352         620 LC     6              C[M] = 04096.....
   295  353         112 C=0    WPT            CLEAR REGISTER C[7:0]
   296  354          12 A=0    WPT            CLEAR REGISTER A[7:0]
   297  355          34 PT=    3              POINT AT LOWEST 2 BYTES
   298  356         302 C=B    PT             GET HIGH DIGIT OF ORIG #
   299  357        1240 SETDEC                ENTER DECIMAL MODE
   300  360 B-D20  1142 C=C-1  PT             SUBTRACT 1 FROM HIGH DIGIT
   301  361          37 GOC    B-D30  ( 364) EXIT WHEN -1 IS REACHED
   302  362         532 A=A+C  M              ADD 4096 FOR EACH HEX 1000
   303  363        1753 GONC   B-D20  ( 360) KEEP DECR. HIGH DIGIT
   304  364 B-D30  1140 SETHEX                ENTER HEXADECIMAL MODE
   305  365        1604 S0=    0              INIT LEADING ZERO FLAG
   306  366 PRBI10  216 B=A                   SAVE 5-DIGIT DECIMAL TO B
   307  367         460 LDI                   LOAD LOW 12 BITS OF C WITH
   308  370          40 CON    @40            @40 = ASCII BLANK
   309  371        1634 PT=    0              POINT AT LOWEST BYTE OF C
   310  372         130 G=C                   SAVE ASCII BLANK TO G
   311  373        1534 PT=    12             POINT AT HIGHEST BYTE OF C
   312  374        1302 ? B#0  PT             IS HIGHEST DIGIT A ZERO ?
   313  375          37 GOC    PRBI40 ( 400) NO, IT IS DIGIT 1 TO 9
   314  376        1614 ?S0=1                 IS IT A LEADING ZERO ?
   315  377          63 GONC   PRBI45 ( 405) YES, OUTPUT BLANK INSTEAD
   316  400 PRBI40 1610 S0=    1              SET WHEN DIGIT ENCOUNTERED
   317  401        1334 PT=    13             POINT AT SIGN DIGIT OF C
   318  402         316 C=B                   GET 5-DIGIT DECIMAL FROM B
   319  403         320 LC     3              CHANGE HIGH DIGIT TO ASCII
```

```
   320  404          130 G=C                  SAVE ASCII BYTE TO REG G
   321  405 PRBI45     1 GOSUB  APPEND        APPEND CHAR IN G TO ALPHA
   321  406            0                      *MAINFRAME: CN11, @0416
   322  407          156 AB EX                MOVE 5-DIGIT DECIMAL TO A
   323  410         1772 A SL    M            SHIFT OUT HIGHEST DIGIT
   324  411          676 A=A-1   S            OUTPUT 5 DIGITS ALREADY
   325  412         1543 GONC    PRBI10 ( 366) NOT YET, KEEP OUTPUTTING
   326  413            1 GOSUB  PRT11?         SEE IF NEEDS TO PRINT ?
   326  414            0                      *ILCAS&CTL:  CS1, @0424
   327  415            1 GOSUB  FNDCAS         LOOK FOR CASSETTE AGAIN
   327  416            0                      *ILCAS&CTL:  CS0, @0566
   328  417            0 NOP                  P+1 - FALL THROUGH TO
   329  420            1 GOSUB  PCHKKB         P+2 - CHECK KEYBOARD
   329  421            0                      *ILCAS&CTL:  CS3, @1470
   330  422            1 GOLONG DIR150         READ NEXT FILE ENTRY
   330  423            2                      *ILCAS&CTL:  CS1, @0133
*
   332                   ENTRY  PRT11?
*
   334  424 PRT11?     1 GOSUB  LDSST0         SEE IF MANUAL MODE SET
   334  425            0                      *MAINFRAME:  CN1, @1627
   335  426          274 RCR    5             C[1:0] = FLAGS 28-35
   336  427         1730 CST EX               MOVE C[1:0] TO STATUS
   337  430           14 ?S3=1                 IN MANUAL MODE (FLAG 32) ?
   338  431         1540 RTN C                 YES, DON'T PRINT
   339  432         1730 CST EX               RESTORE STATUS FLAGS
   340  433         1614 ?S0=1                 PRINTER PRESENT ?
   341  434         1640 RTN NC                NO, DON'T PRINT
   342  435         1545 CON    @1545          GOSUB PRT11
   343  436          674 CON    @674           *ILPRINTER:  PL3, @1731
   344  437            1 GOLONG FLSHRT         RTN ADDR OF "NFRPU" TO STK
   344  440            2                      *ILCAS&CTL:  CS2, @0173
   345                   FILLTO @442
        441         0000 NOP
        442         0000 NOP
*
*
*************************************************************************
* SEC   - SECURE A FILE                                                 *
* UNSEC - UNSECURE A FILE                                               *
*************************************************************************
*
   353                   ENTRY  SEC
   354                   ENTRY  UNSEC
*
   356  443          203 CON    @203           C
   357  444            5 CON    @05            E
   358  445           23 CON    @23            S
   359  446 SEC        1 GOSUB  FLSCHJ         SEARCH FOR THE FILE
   359  447            0                      *ILCAS&CTL:  CS2, @0005
   360  450          410 S8=    1              S8=1 TO SECURE THE FILE
   361  451          113 GOTO   SEQ10  ( 462) CONTINUE TO SHARED CODE
*
   363  452          203 CON    @203           C
   364  453            5 CON    @05            E
   365  454           23 CON    @23            S
   366  455           16 CON    @16            N
   367  456           25 CON    @25            U
   368  457 UNSEC      1 GOSUB  FLSCHJ         SEARCH FOR THE FILE
   368  460            0                      *ILCAS&CTL:  CS2, @0005
```

```
   369  461          404 S8=    0             S8=0 TO UNSECURE THE FILE
   370  462 SEQ10    260 C=N                  GET FILE INFO FROM N
   371  463         1574 RCR    12            C[1:0] = FILE TYPE
   372  464         1530 ST=C                 MOVE FILE TYPE TO STATUS
   373  465            4 S3=    0             ASSUME IT IS UNPROTECTED
   374  466          414 ?S8=1                 SECURING THE FILE ?
   375  467           23 GONC   *+2    ( 471) NO, SKIP NEXT 2 LINES
   376  470           10 S3=    1             MAKE FILE TYPE PROTECTED
   377  471         1630 C=ST                 PUT STATUS BACK TO C[1:0]
   378  472         1074 RCR    2             ROTATE BACK INTO POSITION
   379  473          160 N=C                  REWRITE INFORMATION TO N
   380  474         1076 C=C+1  S             IS IT A 41C FILE ?
   381  475            1 GOLC   FLTYER        NO, DON'T DO SECURE THEN
   381  476            3                      *ILCAS&CTL:  CS2, @0212
   382  477          333 GOTO   RNAM10 ( 532) RESEND THE FILE ENTRY
*
*************************************************************************
* RENAME - RENAME A FILE                                                *
*************************************************************************
*
   388                 ENTRY  RENAME
   389                 ENTRY  RNAM10
*
   391  500          205 CON    @205          E
   392  501           15 CON    @15           M
   393  502            1 CON    @01           A
   394  503           16 CON    @16           N
   395  504            5 CON    @05           E
   396  505           22 CON    @22           R
   397  506 RENAME 1010 S2=     1             SKIP OVER OLD NAME IN A-REG
   398  507            1 GOSUB  AOUT1         GET THE NEW NAME INTO M
   398  510            0                      *ILCAS&CTL:  CS0, @1064
   399  511           76 B=0    S             CLEAR SIGN DIGIT OF B
   400  512            1 GOSUB  FLSCHX        CHECK DUPLICATE FILE NAME
   400  513            0                      *ILCAS&CTL:  CS2, @0016
   401  514          630 C=M                  GET THE FILE NAME FROM M
   402  515         1356 ? C#0                FOUND THE NAME ?
   403  516            1 GOLC   DUPFL         YES, DISPLAY "DUP FL NAME"
   403  517            3                      *ILCAS&CTL:  CS1, @1222
   404  520            1 GOSUB  FLSCHJ        SEARCH FOR THE FILE
   404  521            0                      *ILCAS&CTL:  CS2, @0005
   405  522            1 GOSUB  CHKPCT        CHECK IF FILE IS PROTECTED
   405  523            0                      *ILCAS&CTL:  CS3, @0414
   406  524         1010 S2=    1             LAST CHAR ADR IN R8[13:10]
   407  525            4 S3=    0             SHIFT FILE INTO M F/LEFT
   408  526            1 GOSUB  AOUTFL        GET OLD FILE NAME TO M
   408  527            0                      *ILCAS&CTL:  CS0, @1066
   409  530            1 GOSUB  AOUTFL        GET NEW FILE NAME TO M
   409  531            0                      *ILCAS&CTL:  CS0, @1066
   410  532 RNAM10    76 B=0    S             DO COPYBF (COPY BUFFER)
   411  533            1 GOLONG REWENT        REWRITE THE FILE ENTRY
   411  534            2                      *ILCAS&CTL:  CS3, @0145
*
*************************************************************************
* WRTRX  - WRITE REGISTERS SEQUENTIALLY AS DIRECTED BY THE X-REGISTER   *
*          WRITE A REGISTER BLOCK TO A DATA FILE STARTING FROM THE      *
*          LOCATION OF THE FILE POINTER USING X-REGISTER AS A GUIDE     *
* INPUT:   X-REG = BBB.EEE                                              *
*            WHERE BBB = STARTING REGISTER NUMBER                       *
*                  EEE = ENDING REGISTER NUMBER                         *
```

```
*          NO FILE NAME IS REQUIRED, WILL VERIFY THAT THE FILE         *
*          POINTED TO IS A DATA FILE BEFORE WRITING                    *
*********************************************************************
*
   424                   ENTRY  WRTRX
*
   426  535          230 CON    @230          X
   427  536           22 CON    @22           R
   428  537           24 CON    @24           T
   429  540           22 CON    @22           R
   430  541           27 CON    @27           W
   431  542 WRTRX      1 GOSUB  CHKCST        SEE IF CASSETTE IS READY
   431  543            0                      *ILCAS&CTL:  CS3, @0335
   432  544            1 GOSUB  SVBREN        SAVE BYTE # & READ FL ENTRY
   432  545            0                      *ILCAS&CTL:  CS3, @1317
   433  546            1 GOSUB  FLSHDT        SEE IF IT IS A DATA FILE
   433  547            0                      *ILCAS&CTL:  CS2, @0203
   434  550            1 GOSUB  CHKPCT        CHECK IF FILE IS PROTECTED
   434  551            0                      *ILCAS&CTL:  CS3, @0414
   435  552            1 GOSUB  DFBDCK        CHECK FILE BOUNDARY
   435  553            0                      *ILCAS&CTL:  CS3, @1140
   436  554          230 C=G                  GET BYTE RTND FROM SVBREN
   437  555         1342 ? C#0  PT             WAS LAST TIME A WRITE ?
   438  556          177 GOC    WRRS10 ( 575) YES, JUST CONTINUE TO WRITE
   439  557            1 GOSUB  SVMODE        SAVE MODE IN BYTE 253
   439  560            0                      *ILCAS&CTL:  CS3, @1417
   440  561         1170 C=REGN 9              BACK UP ONE RECORD
   441  562         1074 RCR    2              C[X] = # OF RECS PAST BOF
   442  563          416 A=C                   SAVE # OF RECS IN REG A
   443  564          260 C=N                   GET FILE INFO FROM REG N
   444  565          474 RCR    8              C[X] = STARTING RECORD #
   445  566          506 A=A+C  X              A[X] = CURRENT RECORD #
   446                   LEGAL                 (CLEAR THE CARRY FLAG)
   447  567            1 GOSUB  SEEK          LOOK FOR CURRENT RECORD
   447  570            0                      *ILCAS&CTL:  CS3, @1565
   448  571            1 GOSUB  PARWRT        SEND CMD - PARTIAL WRITE
   448  572            0                      *ILCAS&CTL:  CS1, @0026
   449  573            1 GOSUB  RSTBP         RESTORE BYTE POINTER
   449  574            0                      *ILCAS&CTL:  CS3, @1644
   450  575 WRRS10   233 GOTO   SNDRG0 ( 620) GOTO WRITE THE REGISTERS
*
*********************************************************************
* WRTR  -  WRITE REGISTERS                                          *
*          WRITE A BLOCK OF COCONUT REGISTERS TO A FILE              *
*          STARTING FROM THE BEGINNING OF THE FILE                   *
* INPUT:   ALPHA REG - FILE NAME                                     *
*                                                                    *
* SNDRG0 - SPECIAL ENTRY                                             *
* INPUT:   M[X]   = NUMBER OF REGISTERS TO WRITE                     *
*          M[5:3] = STARTING REGISTER ADDRESS                        *
*********************************************************************
*
   463                   ENTRY  WRTR
   464                   ENTRY  WRTRXX
*
   466  576          222 CON    @222          R
   467  577           24 CON    @24           T
   468  600           22 CON    @22           R
   469  601           27 CON    @27           W
   470  602 WRTR      1 GOSUB  FLSCHD        SEARCH THE DATA FILE
```

```
   470  603           0                        *ILCAS&CTL:  CS2, @0010
   471  604           1 GOSUB  CHKPCT          CHECK IF FILE IS PROTECTED
   471  605           0                        *ILCAS&CTL:  CS3, @0414
   472  606           1 GOSUB  SVENTW          SAVE CURRENT FILE ENTRY
   472  607           0                        *ILCAS&CTL:  CS3, @1365
   473  610           1 GOSUB  DATALL          CHECK REG & FILE SIZE
   473  611           0                        *ILCAS&CTL:  CS3, @1301
   474  612 WRTRXX    1 GOSUB  SEEKN           SEEK TO STARTING RECORD
   474  613           0                        *ILCAS&CTL:  CS3, @1562
*
   476                  ENTRY  SNDRGS
   477                  ENTRY  SNDRGA
   478                  ENTRY  SNDR10
   479                  ENTRY  SNDRG0
   480                  ENTRY  SNDRDN
   481                  ENTRY  CSCKUT
*
   483  614           1 GOSUB  PARWRT          SEND SEC.CMD - PARTIAL WRT
   483  615           0                        *ILCAS&CTL:  CS1, @0026
   484  616           1 GOSUB  LISTEN          ADDRESS DEVICE AS LISTENER
   484  617           0                        *ILCAS&CTL:  CS0, @0335
   485  620 SNDRG0   404 S8=    0              NOT CALLED BY WRITE ALL
   486  621 SNDRGS     1 GOSUB  DTFLOW          SEND SEC.CMD - DATA FOLLOW
   486  622           0                        *ILCAS&CTL:  CS0, @0332
   487  623 SNDRGA   144 HPL=CH 1              WRITE CONTROL INT REGISTER
   488  624           5 CH=    @001            ENABLE FI LINE (BIT 0)
   489  625 SNDR10   630 C=M                    GET REG ADDR & REG COUNT
   490  626        1146 C=C-1  X                ALL DONE ?
   491  627         177 GOC    SNDR30 ( 646)   YES, FINISH WRITING REGS
   492  630          74 RCR    3               C[X] = TARGET REG ADDR
   493  631        1160 DADD=C                   POINT TO TARGET REGISTER
   494  632        1046 C=C+1  X                POINT TO NEXT REGISTER
   495  633         674 RCR    11              ROTATE BACK INTO POSITION
   496  634         530 M=C                     PUT REG ADDR & REG COUNT
   497  635          70 C=DATA                  READ THE TARGET REGISTER
   498  636         416 A=C                     COPY TO REGISTER A
   499  637         456 A=A+B                   UPDATE THE CHECKSUM
   500  640         216 B=A                     RETURN CHECKSUM TO B
   501  641           1 GOSUB  SNDRGC          SEND CONTENTS OF REG C
   501  642           0                        *ILCAS&CTL:  CS1, @0717
   502  643        1114 ?S9=1                   ANY ERRORS ?
   503  644        1613 GONC   SNDR10 ( 625)   NO, GET NEXT REGISTER
   504  645         133 GOTO   WRERCK ( 660)   YES, GOTO ERROR CHECK
   505  646 SNDR30   414 ?S8=1                  CALLED BY WRITE ALL ?
   506  647        1540 RTN C                   NO, WE'RE DONE HERE
   507  650 SNDRDN   460 LDI                    SEND CLOSE RECORD CMD
   508  651         250 CON    @250            @250 = DDL 8 COMMAND
   509  652           1 GOSUB  SCMD            SEND DDL 8 (CLOSE RECORD)
   509  653           0                        *ILCAS&CTL:  CS0, @0272
   510  654 CSCKUT    1 GOSUB  WAITS           WAIT FOR DONE, CHECK ERROR
   510  655           0                        *ILCAS&CTL:  CS1, @0032
   511  656           1 GOLONG UNT             SEND UNTALK COMMAND
   511  657           2                        *ILCAS&CTL:  CS0, @0254
   512  660 WRERCK    1 GOLONG CSERCK          CASSETTE ERROR CHECK
   512  661           2                        *ILCAS&CTL:  CS1, @1064
*
*****************************************************************
* ZERO   - WRITE ZEROS TO A DATA FILE                           *
*****************************************************************
* ZEROFL - ROUTINE TO WRITE ZEROS TO A FILE                     *
```

```
* IN&OUT:  N[7:4] = # OF RECORDS                                        *
*          N[3:0] = STARTING RECORD #                                   *
*************************************************************************
   521                   ENTRY  ZERO
   522                   ENTRY  ZEROFL
*
   524  662          217 CON    @217           O
   525  663           22 CON    @22            R
   526  664            5 CON    @05            E
   527  665           32 CON    @32            Z
   528  666 ZERO       1 GOSUB  FLSCHD         SEARCH THE DATA FILE
   528  667            0                       *ILCAS&CTL:  CS2, @0010
   529  670            1 GOSUB  CHKPCT         CHECK IF FILE IS PROTECTED
   529  671            0                       *ILCAS&CTL:  CS3, @0414
   530  672         1610 S0=    1              REMEMBER LAST OPERATION
   531  673            1 GOSUB  SVMODE          WAS A WRITE
   531  674            0                       *ILCAS&CTL:  CS3, @1417
   532  675 ZEROFL     1 GOSUB  SEKSUB         SEEK TO THE FILE
   532  676            0                       *ILCAS&CTL:  CS0, @0326
   533  677          260 C=N                   GET FILE INFO FROM REG N
   534  700          174 RCR    4              C[X] = # OF RECS IN FILE
   535  701         1434 PT=    1              POINT AT LOWEST BYTE OF C
   536  702 ZERO10  1146 C=C-1  X              DONE WITH ALL RECORDS ?
   537  703         1517 GOC    CSCKUT ( 654) YES, CHK ERRORS AND EXIT
   538  704          160 N=C                   REPLACE FILE INFO IN N
   539  705            6 A=0    X              CLEAR BYTE COUNT TOTAL
   540  706 ZERO20   106 C=0    X              BYTE TO SEND IS 00
   541  707            1 GOSUB  SDATA0         SEND A BYTE OF DATA
   541  710            0                       *ILCAS&CTL:  CS0, @0446
   542  711         1114 ?S9=1                 ANY ERRORS ?
   543  712         1467 GOC    WRERCK ( 660) YES, CHECK ERRORS & EXIT
   544  713          552 A=A+1  WPT            WRITE 256 BYTES YET
   545  714         1723 GONC   ZERO20 ( 706) NOT YET, WRITE MORE ZEROS
   546  715          260 C=N                   GET FILE INFO FROM REG N
   547  716         1643 GOTO   ZERO10 ( 702) WRITE THE NEXT RECORD
*************************************************************************
* SNDRGC - SEND THE CONTENTS OF REGISTER C                              *
* INPUT:   CASSETTE IS IN THE MIDDLE OF RECEIVING DATA                  *
* USES:    A, C, +2 SUBROUTINE LEVELS                                   *
*************************************************************************
   553                   ENTRY  SNDRGC
*
   555  717 SNDRGC   416 A=C                   SAVE COPY OF REGISTER C
   556  720            1 GOSUB  SDATA          SEND E1 & E2 (DIGIT 1 & 0)
   556  721            0                       *ILCAS&CTL:  CS0, @0450
   557  722          256 AC EX                 RESTORE SAVED DATA FROM A
   558  723         1706 C SR   X              SHIFT OUT E2 (DIGIT 0)
   559  724         1706 C SR   X              SHIFT OUT E1 (DIGIT 1)
   560  725         1374 RCR    13             C[1] = E0, C[0] = MS
   561  726          416 A=C                   SAVE COPY OF REGISTER C
   562  727            1 GOSUB  SDATA          SEND E0 & MS
   562  730            0                       *ILCAS&CTL:  CS0, @0450
   563  731         1634 PT=    0              SEND 00 FOLLOWED BY M9-M0
   564  732 SNDR20   256 AC EX                 RESTORE SAVED DATA FROM A
   565  733         1074 RCR    2              ROTATE RIGHT ONE BYTE
   566  734          416 A=C                   SAVE COPY OF REGISTER C
   567  735            1 GOSUB  SDATA          SEND ANOTHER BYTE OF DATA
   567  736            0                       *ILCAS&CTL:  CS0, @0450
   568  737         1734 INC PT                MOVE TO THE NEXT DIGIT
   569  740          524 ? PT=  6              LAST BYTE SENT ?
```

```
   570  741        1540 RTN C                YES, RTN FROM SUBROUTINE
   571  742        1114 ?S9=1                ANY ERRORS ?
   572  743        1673 GONC   SNDR20 ( 732) NO, GO GET NEXT BYTE
   573  744        1740 RTN                  YES, RTN FROM SUBROUTINE
*
*************************************************************************
* READRX - READ REGISTERS SEQUENTIALLY AS DIRECTED BY THE X-REGISTER   *
*          READ A DATA FILE STARTING FROM THE FILE POINTER LOCATION    *
*          AND WRITE TO 41C DATA REGISTERS USING X-REGISTER AS GUIDE   *
*                                                                      *
* INPUT:   NO FILE NAME IS REQUIRED, THE FILE HAS TO BE VERIFIED AS A  *
*          DATA FILE.                                                  *
*          X-REG = BBB.EEE (BBB = BEGINNING, EEE = ENDING)             *
*************************************************************************
*
   585                  ENTRY  READRX
*
   587  745         230 CON    @230          X
   588  746          22 CON    @22           R
   589  747           4 CON    @04           D
   590  750           1 CON    @01           A
   591  751           5 CON    @05           E
   592  752          22 CON    @22           R
   593  753 READRX    1 GOSUB  CHKCST        SEE IF CASSETTE IS READY
   593  754           0                       *ILCAS&CTL:  CS3, @0335
   594  755           1 GOSUB  SVBREN         SAVE BYTE # & READ ENTRY
   594  756           0                       *ILCAS&CTL:  CS3, @1317
   595  757           1 GOSUB  FLSHDT         SEE IF IT IS A DATA FILE
   595  760           0                       *ILCAS&CTL:  CS2, @0203
   596  761           1 GOSUB  DFBDCK         CHECK FILE BOUNDARY
   596  762           0                       *ILCAS&CTL:  CS3, @1140
   597  763         230 C=G                   GET BYTE RTND FROM SVBREN
   598  764        1342 ? C#0  PT             WAS LAST TIME A WRITE ?
   599  765         123 GONC   RDRS10 ( 777)  NO, LAST TIME WAS A READ
   600  766        1604 S0=    0              REMEMBER WAS READ LAST TIME
   601  767           1 GOSUB  SVMODE         SAVE MODE
   601  770           0                       *ILCAS&CTL:  CS3, @1417
   602  771           1 GOSUB  TALKER         ADDRESS DEVICE AS A TALKER
   602  772           0                       *ILCAS&CTL:  CS0, @0262
   603  773           1 GOSUB  SEEK40         YES, DO A READ
   603  774           0                       *ILCAS&CTL:  CS0, @1640
   604  775           1 GOSUB  RSTBP          PUT BYTE # BACK
   604  776           0                       *ILCAS&CTL:  CS3, @1644
   605  777 RDRS10    1 GOSUB  TALKER         ADDRESS DEVICE AS A TALKER
   605 1000           0                       *ILCAS&CTL:  CS0, @0262
   606 1001         163 GOTO   RDREG0 (1017)  READ REGISTERS ROUTINE
*
*************************************************************************
* READR - READ REGISTERS                                               *
*         READ FROM THE BEGINNING OF A DATA FILE AND STORE TO DATA     *
*         REGISTERS UNTIL ALL REGS ARE FULL OR END OF DATA REACHED.    *
* INPUT:  ALPHA REG - FILE NAME                                        *
*************************************************************************
*
   615                  ENTRY  READR
*
   617 1002         222 CON    @222          R
   618 1003           4 CON    @04           D
   619 1004           1 CON    @01           A
   620 1005           5 CON    @05           E
```

```
   621 1006           22 CON    @22           R
   622 1007 READR      1 GOSUB  FLSCHD        CHECK THE DATA FILE
   622 1010            0                      *ILCAS&CTL:  CS2, @0010
   623 1011            1 GOSUB  SVENTR        SAVE CURRENT FILE ENTRY
   623 1012            0                      *ILCAS&CTL:  CS3, @1363
   624 1013            1 GOSUB  DATALL        COMPUTE # OF REGS TO READ
   624 1014            0                      *ILCAS&CTL:  CS3, @1301
   625 1015            1 GOSUB  SEEKRN        SEEK & READ FIRST RECORD
   625 1016            0                      *ILCAS&CTL:  CS3, @1567
*
   627                   ENTRY  RDREG         READ REGISTERS ROUTINE
   628                   ENTRY  RDREG0
   629                   ENTRY  RDRG10
   630                   ENTRY  CSERCK
*
   632 1017 RDREG0   404 S8=    0             NOT CALLED BY READ ALL
   633 1020 RDREG      1 GOSUB  SNDATA        SEND "SEND DATA" COMMAND
   633 1021            0                      *ILCAS&CTL:  CS0, @0346
   634 1022          630 C=M                  GET REG ADDR & REG COUNT
   635 1023 RDRG10  1146 C=C-1  X             DECREMENT REGISTER COUNT
   636 1024          530 M=C                  PUT REG ADDR & REG COUNT
   637 1025 RDRG15   630 C=M                  GET REG ADDR & REG COUNT
   638 1026           74 RCR    3             C[X] = TARGET REG ADDR
   639 1027         1160 DADD=C               POINT TO TARGET REGISTER
   640 1030         1046 C=C+1  X             POINT TO NEXT REGISTER
   641 1031          674 RCR    11            ROTATE BACK INTO POSITION
   642 1032          530 M=C                  PUT REG ADDR & REG COUNT
   643 1033            1 GOSUB  RDRGA         READ 7 BYTES & PUT INTO A
   643 1034            0                      *ILCAS&CTL:  CS1, @1076
   644 1035         1360 DATA=C               STORE THE REGISTER
   645 1036          456 A=A+B                UPDATE THE CHECKSUM
   646 1037          216 B=A                  RETURN CHECKSUM TO B
   647 1040          674 RCR    11            ROTATE BACK AFTER RDRGA
   648 1041          730 CM EX                GET REG COUNT FROM M
   649 1042         1146 C=C-1  X             LAST REG TO READ ?
   650 1043           47 GOC    RDRG30 (1047) YES, IF CARRY IS SET
   651 1044          730 CM EX                GET C-REGISTER FROM M
   652 1045         1200 HPIL=C 2             ECHO DATA FRAME TO LOOP
   653 1046         1573 GOTO   RDRG15 (1025) READ NEXT FRAME OF DATA
   654 1047 RDRG30   730 CM EX                GET LAST DATA BYTE BACK
   655 1050          414 ?S8=1                CALLED BY READ ALL ?
   656 1051           33 GONC   RDRGDN (1054) NO, FINISH UP READ REGS
   657 1052         1200 HPIL=C 2             ECHO DATA FRAME TO LOOP
   658 1053         1740 RTN                  END OF READ REGISTERS
   659 1054 RDRGDN     1 GOSUB  NRDC          SEND "NOT READY" COMMAND
   659 1055            0                      *ILCAS&CTL:  CS0, @0366
   660 1056         1114 ?S9=1                ANY ERRORS ?
   661 1057           57 GOC    CSERCK (1064) YES, SEE WHAT ERROR
   662 1060            1 GOLONG UNT           SEND UNTALK AND EXIT
   662 1061            2                      *ILCAS&CTL,  CS0, @0254
   663 1062 RDRG40   414 ?S8=1                CALLED BY READ ALL ?
   664 1063         1540 RTN C                YES, JUST RETURN
   665 1064 CSERCK  1104 S9=    0             CLEAR ERROR FLAG
   666 1065            1 GOSUB  CSSTAS        READ CASSETTE STATUS
   666 1066            0                      *ILCAS&CTL:  CS0, @0700
   667 1067         1114 ?S9=1                CSSTAS RETURNED AN ERROR ?
   668 1070           27 GOC    LOPERR (1072) LOOP TRANSMIT ERROR
   669 1071          114 ?S4=1                ANY CASSETTE ERROR ?
   670 1072 LOPERR     1 GOLNC  PILERR        NO, MUST BE PIL XMIT ERROR
   670 1073            2                      *ILCAS&CTL:  CS1, @1751
```

```
   671 1074           1 GOLONG CSERR         DETERMINE CASSETTE ERROR
   671 1075           2                      *ILCAS&CTL:  CS1, @0042
*****************************************************************
* RDRGA - READ 7 DATA FRAMES AND SAVE TO REGISTER A             *
* INPUT:  CASSETTE IS IN THE MIDDLE OF READING DATA             *
* USES:   A, C, PT, +1 SUBROUTINE LEVEL                         *
*****************************************************************
   677                ENTRY  RDRGA
*
   679 1076 RDRGA     1 GOSUB  RDDFRM        READ E1 AND E2
   679 1077           0                      *ILCAS&CTL:  CS0, @0420
   680 1100        1200 HPIL=C 2             ECHO BYTE READ TO LOOP
   681 1101          74 RCR    3             C[13:11] = 0, E1, E2
   682 1102         416 A=C                  A[13:11] = 0, E1, E2
   683 1103           1 GOSUB  RDDFRM        READ E0 & MS FRAME
   683 1104           0                      *ILCAS&CTL:  CS0, @0420
   684 1105        1200 HPIL=C 2             ECHO BYTE READ TO LOOP
   685 1106        1074 RCR    2             C[13] = E0
   686 1107         436 A=C    S             A[S] = E0
   687 1110        1074 RCR    2             C[10] = MS
   688 1111         334 PT=    10            POINT AT MS DIGIT
   689 1112         242 AC EX  PT            A = E0, E1, E2, MS
   690 1113           1 GOSUB  RDDFRM        READ A DUMMY BYTE
   690 1114           0                      *ILCAS&CTL:  CS0, @0420
   691 1115        1200 HPIL=C 2             ECHO BYTE READ TO LOOP
   692 1116 RDRG50    1 GOSUB  RDDFRM        READ M9-M0
   692 1117           0                      *ILCAS&CTL:  CS0, @0420
   693 1120        1114 ?S9=1                ANY ERRORS ?
   694 1121        1417 GOC    RDRG40 (1062) YES, CHECK STATUS & EXIT
   695 1122         256 AC EX                RESTORE SAVED DATA FROM A
   696 1123         246 AC EX  X             SWAP A & C LOWEST BYTE
   697 1124         266 AC EX  XS            SWAP A & C EXPONENT SIGN
   698 1125        1724 DEC PT               MOVE TO NEXT DIGIT
   699 1126         224 ? PT=  5             LAST BYTE READ ?
   700 1127          57 GOC    RDRG20 (1134) YES, CLEAN UP & EXIT
   701 1130        1200 HPIL=C 2             ECHO BYTE READ TO LOOP
   702 1131        1074 RCR    2             ROTATE RIGHT ONE BYTE
   703 1132         416 A=C                  SAVE COPY OF REGISTER C
   704 1133        1633 GOTO   RDRG50 (1116) READ ANOTHER BYTE OF DATA
   705 1134 RDRG20   74 RCR    3             FINISH ROTATING REG C
   706 1135         416 A=C                  SAVE COPY OF REGISTER C
   707 1136        1740 RTN                  END OF READING REGISTERS
*
*****************************************************************
* SEEKR  - SET THE FILE POINTER IN A DATA FILE TO A GIVEN REGISTER*
*          NUMBER, SO THE FOLLOWING READ OR WRITE REGISTERS     *
*          SEQUENTIALLY CAN START FROM THE POINTER (INCLUSIVE). *
* EXAMPLE: SEEK TO REGISTER 23 WILL MAKE THE NEXT READ OR WRITE *
*          STARTING FROM REGISTER 23                            *
* INPUT:   ALPHA REGISTER = FILE NAME                           *
*          X-REGISTER     = DESTINATION REGISTER NUMBER         *
*****************************************************************
*
   719                ENTRY  SEEKR
*
   721 1137         222 CON    @222          R
   722 1140          13 CON    @13           K
   723 1141           5 CON    @05           E
   724 1142           5 CON    @05           E
   725 1143          23 CON    @23           S
```

```
   726 1144 SEEKR     1 GOSUB  FLSCHD        SEARCH THE DATA FILE
   726 1145           0                      *ILCAS&CTL:  CS2, @0010
   727 1146           1 GOSUB  SVENTR        SAVE CURRENT FILE ENTRY
   727 1147           0                      *ILCAS&CTL:  CS3, @1363
   728 1150           1 GOSUB  X-BIN         CONVERT X TO BINARY
   728 1151           0                      *ILCAS&CTL:  CS1, @1612
   729 1152           1 GOSUB  RG-BY#        MUTIPLY REG # BY 7
   729 1153           0                      *ILCAS&CTL:  CS2, @1761
   730 1154        1074 RCR    2             ROTATE C RIGHT ONE BYTE
   731 1155         416 A=C                  A[3:0] = DESTINATION REG
   732 1156         260 C=N                  C[3:0] = # REGS AVAILABLE
   733 1157        1156 C=C-1                1ST REG IS REG.0
   734 1160          34 PT=    3             POINT AT LOWEST 2 BYTES
   735 1161         252 AC EX  WPT           A[3:0] = # REGS AVAILABLE
   736 1162        1412 ? A<C  WPT           # REGS AVAIL < DEST REG ?
   737 1163           1 GOLC   CSEOF         YES, CROSSED END OF FILE
   737 1164           3                      *ILCAS&CTL:  CS1, @1474
   738 1165         256 AC EX                C[3:0] = # REGS AVAILABLE
   739 1166         174 RCR    4             C[4:0] = # OF BYTES (DEST)
   740 1167         134 PT=    4             POINT AT LOWEST 2 BYTES
   741 1170         412 A=C    WPT           A[4:0] = # OF BYTES (DEST)
   742 1171         206 B=A    X             SAVE # OF BYTES IN B[X]
   743 1172        1616 A SR                 SHIFT REG A 1 DIGIT RIGHT
   744 1173        1616 A SR                 A[X] = # OF RECS PAST BOF
   745 1174         260 C=N                  GET FILE INFO FROM REG N
   746 1175         474 RCR    8             C[X] = STARTING RECORD #
   747 1176         506 A=A+C  X             A[X] = DESTINATION REC #
   748                LEGAL                  (CLEAR THE CARRY FLAG)
   749 1177           1 GOSUB  SEEKRD        SEEK TO RECORD & READ IT
   749 1200           0                      *ILCAS&CTL:  CS3, @1572
   750 1201         146 AB EX  X             A[X] = NUMBER OF BYTES
   751 1202           1 GOSUB  SETBPL        SET BYTE POINTER
   751 1203           0                      *ILCAS&CTL:  CS3, @1627
   752 1204           1 GOLONG UNL           SEND UNLISTEN COMMAND
   752 1205           2                      *ILCAS&CTL:  CS0, @0257
*
*****************************************************************************
* CREATF - CREATE A DATA FILE ON CASSETTE (USER FUNCTION IS "CREATE")  *
*****************************************************************************
*                                                                           *
* INPUT:                                                                    *
*   INT(X)    - FILE SIZE IN NUMBER OF REGISTERS AND < 99999                *
*   ALPHA REG - FILE NAME (ERROR IF DUPLICATE FILE NAME IS SPECIFIED)  *
*****************************************************************************
   762                ENTRY  CREATF
   763                ENTRY  DUPFL
*
   765 1206         205 CON    @205          E
   766 1207          24 CON    @24           T
   767 1210           1 CON    @01           A
   768 1211           5 CON    @05           E
   769 1212          22 CON    @22           R
   770 1213           3 CON    @03           C
   771 1214 CREATF  136 C=0    S             SEARCH FOR DUPLICATE FILE
   772 1215           1 GOSUB  FLSCH         SEARCH THE FILE
   772 1216           0                      *ILCAS&CTL:  CS2, @0013
   773 1217         630 C=M                  C = FILE NAME FROM FLSCH
   774 1220        1356 ? C#0                DUPLICATE FILE NAME ?
   775 1221         203 GONC   CRT10  (1241) NO, GO CREATE THE FILE
   776 1222 DUPFL     1 GOSUB  PLEREX        DISPLAY "DUP FL NAME"
```

```
   776 1223           0                       *ILCAS&CTL:  CS1, @1663
   777 1224           4 CON    @04            D
   778 1225          25 CON    @25            U
   779 1226          20 CON    @20            P
   780 1227          40 CON    @40                 BLANK
   781 1230           6 CON    @06            F
   782 1231          14 CON    @14            L
   783 1232          40 CON    @40                 BLANK
   784 1233          16 CON    @16            N
   785 1234           1 CON    @01            A
   786 1235          15 CON    @15            M
   787 1236        1005 CON    @1005          E
   788 1237           1 GOLONG CSEREX         CASSETTE ERROR EXIT
   788 1240           2                       *ILCAS&CTL:  CS3, @0376
   789 1241 CRT10     1 GOSUB  CRTFLX         CREATE A FILE USING X
   789 1242           0                       *ILCAS&CTL:  CS1, @1245
   790 1243           1 GOLONG ZEROFL         WRITE ZEROS TO THE FILE
   790 1244           2                       *ILCAS&CTL:  CS1, @0675
*************************************************************************
* CRTFL  - CREATE A FILE                                                *
*                                                                       *
* INPUT:   ALPHA REG - FILE NAME                                        *
*          C[10:6]   - FILE LENGTH IN BYTES                             *
*          C[5:2]    - FILE SIZE IN # OF REGS OR # OF BYTES             *
*          C[1:0]    - FILE TYPE                                        *
* CRTFLX - SAME AS CRTFL EXCEPT THE FILE SIZE IS SPECIFIED IN X         *
*                                                                       *
* OUTPUT:  M         = FILE NAME IN ASCII                               *
*          N[13:12]  = FILE TYPE                                        *
*          N[11:8]   = USER LEVEL FILE SIZE (# OF REGS OR BYTES)        *
*          N[7:4]    = FILE SIZE IN NUMBER OF RECORDS                   *
*          N[3:0]    = STARTING RECORD NUMBER                           *
* USES:    A, B, C, N, S[7:3], +2 SUBROUTINE LEVELS                     *
*          REG.9[13] & REG.9[3:0]                                       *
*************************************************************************
   808                 ENTRY  CRTFL
   809                 ENTRY  CRTFL0
   810                 ENTRY  CRTFLX
   811                 ENTRY  CRF2ND
*
   813 1245 CRTFLX    1 GOSUB  X-BIN          CONVERT X TO BINARY
   813 1246           0                       *ILCAS&CTL:  CS1, @1612
   814 1247        1352 ? C#0  WPT            X = 0 ?
   815 1250           1 GOLNC  X-BER          YES, CAN'T CREATE 0 REGS
   815 1251           2                       *ILCAS&CTL:  CS1, @1646
   816 1252           1 GOSUB  RG-BY#         COMPUTE NUMBER OF BYTES
   816 1253           0                       *ILCAS&CTL:  CS2, @1761
   817 1254        1434 PT=    1              POINT TO FILE TYPE DIGIT
   818 1255        1520 LC     13             FIXED LENGTH DATA FILE
   819 1256 CRTFL0 1634 PT=    0              POINT TO FILE PROTECTION
   820 1257         102 C=0    PT             C[1:0] = FILE TYPE
*************************************************************************
* NOW C[10:6] = FILE LENGTH IN NUMBER OF BYTES                          *
*     C[5:2]  = FILE SIZE IN NUMBER OF BYTES OR NUMBER OF REGISTERS     *
*     C[1:0]  = FILE TYPE                                               *
* WE WANT TO MOVE B[13:10] = FILE SIZE IN NUMBER OF RECORDS             *
*                 B[9:4]   = C[5:0]                                     *
*************************************************************************
   828 1260 CRTFL   574 RCR    6              C[4:0] = LENGTH IN BYTES
   829 1261         416 A=C                   A[13:10]=SIZE, A[9:8]=TYPE
```

```
830 1262         1074 RCR    2              C[2:0] = NUMBER OF RECORDS
831 1263         1434 PT=    1              POINT AT LOWEST BYTE
832 1264         1512 ? A#0  WPT            NEED ANY PARTIAL RECORD ?
833 1265           23 GONC   CRF10  (1267)  NO, INTEGRAL # OF RECORDS
834 1266         1056 C=C+1                 YES, INCREMENT # OF RECS
835 1267 CRF10    406 A=C    X              A[X] = # OF RECORDS
836 1270           34 PT=    3              POINT AT LOWEST 2 BYTES
837 1271            2 A=0    PT             ZERO OUT MSD OF LENGTH
838 1272          256 AC EX                 C[13:10]=SIZE, C[9:8]=TYPE
839 1273          174 RCR    4                AND C[3:0] = # OF RECS
840 1274          356 BC EX                 B=REC(4),SZ(4),TYP(2),XXXX
841 1275            1 GOSUB  R5-R6          SAVE 1ST DRIVE ADDR IN R6
841 1276            0                       *ILCAS&CTL:  CS0, @1742
842 1277 CRF2ND   106 C=0    X              CLEAR EXPONENT PART OF C
843 1300         1160 DADD=C                ENABLE CHIP 0
844 1301            1 GOSUB  SEEKRC         SEEK RECORD ZERO (C[X]=0)
844 1302            0                       *ILCAS&CTL:  CS3, @1571
845 1303            1 GOSUB  RDENT          READ 1ST 32 BYTES OF REC.0
845 1304            0                       *ILCAS&CTL:  CS2, @0241
846 1305          260 C=N                   GET FILE INFO FROM REG N
847 1306          174 RCR    4              C[X] = # OF REC. FOR DIR
848 1307           34 PT=    3              POINT AT LOWEST 2 BYTES
849 1310          412 A=C    WPT            A[3:0] = DIR SIZE IN RECS
850 1311         1170 C=REGN 9              GET STARTING REC. #, ETC.
851 1312          252 AC EX  WPT            C[3:0] = DIR SIZE IN RECS
852 1313         1046 C=C+1  X              PARTIALLY FORMATTED TAPE ?
853 1314            1 GOLC   TAPERR         YES (C[X] = FFF IS PFTAPE)
853 1315            3                       *ILCAS&CTL:  CS1, @0056
854 1316         1046 C=C+1  X              C[3:0]=1ST REC.# AFTER DIR
855 1317         1150 REGN=C 9               ASSUMING DIR STARTS AT R2
856 1320         1146 C=C-1  X              SUBTRACT RECORD NUMBER 0
857 1321         1146 C=C-1  X              SUBTRACT RECORD NUMBER 1
858 1322          746 C=C+C  X              DOUBLING C[X] THREE TIMES
859 1323          746 C=C+C  X              MULTIPLIES BY A FACTOR OF
860 1324          746 C=C+C  X              8 ENTRIES PER RECORD
861 1325          406 A=C    X              A[X] = TOTAL # OF ENTRIES
862 1326         1270 C=REGN 10             SAVE THE # IN REGISTER 10
863 1327          246 AC EX  X              C[X] = TOTAL # OF ENTRIES
864 1330         1250 REGN=C 10             WRITE THE # TO REGISTER 10
865 1331            1 GOSUB  SEEKR2         SEEK TO REC.2 AND READ IT
865 1332            0                       *ILCAS&CTL:  CS3, @1557
866 1333           46 B=0    X              EXISTING ENTRIES COUNTER
867 1334 CRF20    404 S8=    0              CLEAR TAPE OVERFLOW FLAG
868 1335          146 AB EX  X              CHK IF ENTRY # >= TOTAL #
869 1336          546 A=A+1  X              INCREMENT WITH NEW ENTRY
870 1337         1270 C=REGN 10             C[X] = TOTAL # OF ENTRIES
871 1340         1406 ? A<C  X              ROOM FOR ANOTHER ENTRY ?
872 1341          253 GONC   TRYNXT (1366)  NO, DIR FULL, TRY NXT DRV
873 1342          146 AB EX  X              YES, SAVE NEW # OF ENTRIES
874 1343            1 GOSUB  RENTPH         READ A FILE ENTRY (PATCH)
874 1344            0                       *ILCAS&CTL:  CS0, @1770
875 1345          630 C=M                   C = FFFF...(IF END OF DIR)
876 1346         1056 C=C+1                 REACH END OF DIRECTORY ?
877 1347          233 GONC   CRF40  (1372)  NOT YET, KEEP READING
878 1350           34 PT=    3              POINT AT LOWEST 2 BYTES
879 1351         1170 C=REGN 9              READ STARTING RECORD #
880 1352          352 BC EX  WPT            B[3:0] = STARTING RECORD #
881 1353          316 C=B                   C[3:0] = STARTING REC #
882 1354          406 A=C    X              A[3:0] = STARTING REC #
883 1355          574 RCR    6              C=TYPE,START #,LENGTH,SIZE
```

```
   884 1356          160 N=C                    CHECK IF ENDING RECORD #
   885 1357          174 RCR     4                WILL GO OVER END OF TAPE
   886 1360          506 A=A+C   X              A[X] = START REC # + SIZE
   887 1361          460 LDI                    LOAD LOW 12 BITS OF C WITH
   888 1362         1001 CON     513            TOTAL # RECS ON A TAPE + 1
   889 1363         1406 ? A<C   X              WILL THIS OVERFLOW TAPE ?
   890 1364          437 GOC     CRF45  (1427)  NOT OVER END OF TAPE YET
   891 1365          410 S8=     1              SET TAPE OVERFLOW FLAG
   892 1366 TRYNXT     1 GOSUB   FNTDEV         LOOK FOR ANOTHER DRIVE
   892 1367            0                        *ILCAS&CTL:  CS2, @1657
   893 1370          453 GOTO    NONXT  (1435)  P+1 - NO OTHER DRIVE FOUND
   894 1371         1063 GOTO    CRF2ND (1277)  P+2 - EXAMINE NEXT DRIVE
*
   896 1372 CRF40    260 C=N                    COMPUTE ENDING REC. ADDR
   897 1373          474 RCR     8              C[3:0] = STARTING REC. #
   898 1374          416 A=C                    A[3:0] = STARTING REC. #
   899 1375         1170 C=REGN  9              C[3:0] = LAST START REC #
   900 1376         1406 ? A<C   X              IS THIS A DUMMY ENTRY ?
   901 1377         1357 GOC     CRF20  (1334)  YES, MOVE TO NEXT ENTRY
   902 1400          260 C=N                    NO, TRY COMPUTATION AGAIN
   903 1401          174 RCR     4              C[3:0] = FILE LEN IN RECS
   904 1402          506 A=A+C   X              A[X]=NEXT FILE START REC#
   905 1403         1170 C=REGN  9              GET PREVIOUS START REC#
   906 1404          246 AC EX   X              PUT IN NEW STARTING REC #
   907 1405         1150 REGN=C  9              WRITE NEW STARTING REC #
   908 1406          260 C=N                    GET FILE INFORMATION AGAIN
   909 1407         1376 ? C#0   S              IS THIS A PURGED FILE ?
   910 1410         1247 GOC     CRF20  (1334)  NO, READ NEXT ENTRY
   911 1411          174 RCR     4              C[X] = # OF RECS THIS FILE
   912 1412          416 A=C                    SAVE FILE INFO TO A
   913 1413          316 C=B                    C=RECS(4),SZ(4),TY(2),XXXX
   914 1414          374 RCR     10             C[X] = # OF RECORDS NEEDED
   915 1415         1406 ? A<C   X              IS EMPTY AREA BIG ENOUGH ?
   916 1416         1167 GOC     CRF20  (1334)  NO, READ NEXT ENTRY
   917 1417          474 RCR     8              DON'T CHANGE OLD LEN(RECS)
   918 1420          416 A=C                    A=XXXX,RECS(4),SZ(4),TY(2)
   919 1421          260 C=N                    GET FILE INFORMATION AGAIN
   920 1422         1574 RCR     12             C[5:0] = FILE SIZE & TYPE
   921 1423          234 PT=     5              POINT AT FILE SIZE & TYPE
   922 1424          252 AC EX   WPT            ONLY CHANGE SIZE & TYPE
   923 1425         1074 RCR     2              ROTATE BACK TO ORIGINAL
   924 1426          160 N=C                    WRITE NEW FILE SIZE & TYPE
   925 1427 CRF45      1 GOSUB   AOUTFL         GET FILE NAME INTO M
   925 1430            0                        *ILCAS&CTL:  CS0, @1066
   926 1431            1 GOSUB   RNAM10         SEND FILE ENTRY
   926 1432            0                        *ILCAS&CTL:  CS1, @0532
   927 1433            1 GOLONG  FLSHRT         PUT NFR ADDR TO RTN STACK
   927 1434            2                        *ILCAS&CTL:  CS2, @0173
   928 1435 NONXT    414 ?S8=1                  DIRECTORY FULL ?
   929 1436          103 GONC    DIRFUL (1446)  YES, DISPLAY "DIR FULL"
   930 1437 CSFULL     1 GOSUB   PLEREX         NO, DISPLAY "MEDM FULL"
   930 1440            0                        *ILCAS&CTL:  CS1, @1663
   931 1441           15 CON     @15            M
   932 1442            5 CON     @05            E
   933 1443            4 CON     @04            D
   934 1444         1015 CON     @1015          M
   935 1445           63 GOTO    DSFULL (1453)  DISPLAY MESSAGE: " FULL"
   936 1446 DIRFUL     1 GOSUB   PLEREX         DISPLAY "DIR FULL"
   936 1447            0                        *ILCAS&CTL:  CS1, @1663
   937 1450            4 CON     @04            D
```

```
   938 1451          11 CON    @11           I
   939 1452        1022 CON    @1022         R
   940 1453 DSFULL    1 GOSUB  MESSL         DISPLAY MESSAGE: " FULL"
   940 1454           0                      *MAINFRAME:  CN1, @1757
   941 1455          40 CON    @40                BLANK
   942 1456           6 CON    @06           F
   943 1457          25 CON    @25           U
   944 1460          14 CON    @14           L
   945 1461        1014 CON    @1014         L
   946 1462         273 GOTO   CSERX1 (1511) CASSETTE ERROR EXIT
*
   948                ENTRY  LJDLY
   949                ENTRY  DSDLY
*
   951 1463 LJDLY   410 S8=    1             PRINT & SET MESSAGE FLAG
   952 1464           1 GOSUB  MSG105        PRINT DISP IN NORM & TRACE
   952 1465           0                      *MAINFRAME:  CN7, @0200
   953 1466 DSDLY   460 LDI                  DELAY .6 SECONDS
   954 1467        1777 CON    @1777         MAXIMUM VALUE IN 10 BITS
   955 1470         746 C=C+C  X             DOUBLE IT TO @3776
   956 1471        1146 C=C-1  X             DECREMENT THE TIMER
   957 1472        1773 GONC   *-1    (1471) KEEP GOING UNTIL ZERO
   958 1473        1740 RTN                  END OF TIME DELAY ROUTINE
*
*
   961                ENTRY  CSEOF
*
   963 1474 CSEOF     1 GOSUB  PLEREX        DISPLAY "END OF FILE"
   963 1475           0                      *ILCAS&CTL:  CS1, @1663
   964 1476           5 CON    @05           E
   965 1477          16 CON    @16           N
   966 1500           4 CON    @04           D
   967 1501          40 CON    @40                BLANK
   968 1502          17 CON    @17           O
   969 1503           6 CON    @06           F
   970 1504          40 CON    @40                BLANK
   971 1505           6 CON    @06           F
   972 1506          11 CON    @11           I
   973 1507          14 CON    @14           L
   974 1510        1005 CON    @1005         E
   975 1511 CSERX1  763 GOTO   CSERX2 (1607) CASSETTE ERROR EXIT
*********************************************************************
* NEWTAP - INITIALIZE A TAPE (FORMAT) (USER FUNCTION IS CALLED "NEWM") *
*********************************************************************
*                                                                   *
* NOTE:                                                             *
*   IT WILL TAKE 3 MINUTES TO FORMAT A TAPE, SO AFTER INITIATING    *
*   THE FORMAT FUNCTION WE WILL NOT WAIT UNTIL THE FORMATTING IS    *
*   DONE BEFORE WE GIVE CONTROL BACK TO THE MAINFRAME.              *
*   IF THE FORMATTING HAS NOT BEEN DONE PROPERLY, WE PRESUME IT     *
*   WILL BE CAUGHT LATER BY SOME OTHER READ OR WRITE FUNCTIONS.     *
*********************************************************************
   987                ENTRY  NEWTAP
   988 1512         215 CON    @215          M
   989 1513          27 CON    @27           W
   990 1514         405 CON    @405          E
   991 1515         416 CON    @416          N
   992 1516 NEWTAP    0 NOP                  NON-PRGMBL 3-DIGIT OPERANDS
   993 1517          32 A=0    M             A[X] = REQUIRED ENTRIES #
   994 1520         546 A=A+1  X             A[X] = # OF ENTRIES + 1
```

```
   995 1521          256 AC EX                 C[X] = # OF ENTRIES + 1
   996 1522          756 C=C+C                 DIVIDE THE NUMBER BY 8
   997 1523         1474 RCR    1              8 ENTRIES PER RECORD
   998 1524         1376 ? C#0  S              MOD OF 8 = 0 ?
   999 1525           23 GONC   NWTP10 (1527)  YES, WAS DIVISIBLE BY 8
  1000 1526         1046 C=C+1  X              NO, NEED ONE MORE RECORD
  1001 1527 NWTP10   406 A=C    X              A[X] = # OF RECS NEEDED
  1002 1530          460 LDI                   LOAD LOW 12 BITS OF C WITH
  1003 1531           71 CON    57             MAXIMUM # OF RECS + 1
  1004 1532         1406 ? A<C  X              X <= 56 ?
  1005 1533            1 GOLNC  ERRDE          NO, SHOW MESSAGE: DATA ERR
  1005 1534            2                       *MAINFRAME: CN10, @0055
  1006 1535          206 B=A    X              X SHOULD BE <= 56 AND > 0
  1007 1536          410 S8=    1              S8=1: CHKCS0 CALLED F/NEWM
  1008 1537            1 GOSUB  CHKCS0         SEE IF CASSETTE IS READY
  1008 1540            0                       *ILCAS&CTL:  CS3, @0336
  1009 1541            1 GOSUB  LISTEN         ADDRESS DEVICE AS LISTENER
  1009 1542            0                       *ILCAS&CTL:  CS0, @0335
  1010 1543          460 LDI                   DEV DEPENDENT LISTENER 5
  1011 1544          245 CON    @245           DDL 5 (FORMAT COMMAND)
  1012 1545            1 GOSUB  SCMDWT         SEND COMMAND - FORMAT
  1012 1546            0                       *ILCAS&CTL:  CS1, @0030
  1013 1547          116 C=0                   CLEAR ACCUMULATOR
  1014 1550          134 PT=    4              POINT TO LOW DIGIT OF C[M]
  1015 1551          220 LC     2              CLEAR 2 RECS, START AT 0
  1016 1552          160 N=C                   MOVE TO REG N FOR ZEROFL
  1017 1553            1 GOSUB  ZEROFL         WRITE ZERO TO REC. 0 & 1
  1017 1554            0                       *ILCAS&CTL:  CS1, @0675
*  WHEN RETURN FROM "ZEROFL", N = 00000000000000
*
  1020 1555            1 GOSUB  SEKSUB         SEEK REC 0 & GO WRITE MODE
  1020 1556            0                       *ILCAS&CTL:  CS0, @0326
  1021 1557          116 C=0                   CLEAR ACCUMULATOR
  1022 1560         1334 PT=    13             WRITE L.I.F. ID TO BYTE 0
  1023 1561         1020 LC     8               AND DIR LENGTH TO REC. 0
  1024 1562          134 PT=    4              POINT TO LOW DIGIT OF C[M]
  1025 1563          220 LC     2              C[4] = 2
  1026 1564          120 LC     1              C[3] = 1
  1027 1565         1334 PT=    13             POINT TO MANTISSA SIGN
  1028 1566            1 GOSUB  SNBYTS         SEND C-REG TWICE (14 BYTE)
  1028 1567            0                       *ILCAS&CTL:  CS3, @1674
  1029 1570          134 PT=    4              POINT TO LOW BYTE OF C[M]
  1030 1571          116 C=0                   CLEAR ACCUMULATOR
  1031 1572            1 GOSUB  SNBYTS         SEND 5 BYTES OF ZERO
  1031 1573            0                       *ILCAS&CTL:  CS3, @1674
  1032 1574          306 C=B    X              GET DIR LENGTH FROM B[X]
  1033 1575          144 HPL=CH 1              WRITE CONTROL INT REGISTER
  1034 1576          405 CH=    @101           END CLASS (BIT 6)
  1035 1577            1 GOSUB  SDATA          SEND OUT AN END FRAME
  1035 1600            0                       *ILCAS&CTL:  CS0, @0450
  1036 1601            1 GOSUB  WAITS          WAIT UNTIL IT IS DONE
  1036 1602            0                       *ILCAS&CTL:  CS1, @0032
  1037 1603            1 GOSUB  INTDIR         INITIALIZE DIR. BUFFER
  1037 1604            0                       *ILCAS&CTL:  CS3, @0361
  1038 1605            1 GOLONG NFRPU          NORMAL FUNC RETURN W/PUSH
  1038 1606            2                       *MAINFRAME:  CN0, @0360
*
  1040 1607 CSERX2     1 GOLONG CSEREX         CASSETTE ERROR EXIT
  1040 1610            2                       *ILCAS&CTL:  CS3, @0376
  1041                   FILLTO @1611
```

```
      1611        0000 NOP
*********************************************************************
* X-BIN - CONVERT THE INTEGER PART OF X-REG TO BINARY                *
*                                                                   *
* INPUT:  NOTHING                                                   *
* OUTPUT: C[3:0] = BINARY OF INT(X)                                 *
* USES:   A, B[X], C, P, Q, +1 SUBROUTINE LEVEL                     *
*********************************************************************
  1049                  ENTRY  X-BIN
  1050                  ENTRY  X-BINC
  1051                  ENTRY  X-BER
*
  1053 1612 X-BIN     1 GOSUB  ACKX          CHECK X-REG FOR NUMBER
  1053 1613           0                      *ILCAS&CTL:  CS0, @1760
  1054 1614 X-BINC  416 A=C                  COPY RESULT TO REG A
  1055 1615        1526 ? A#0  XS            X < 1 ?
  1056 1616          23 GONC   X-BIN1 (1620) NO, DON'T CLEAR REG A
  1057 1617          16 A=0                  YES, CLEAR REGISTER A
  1058 1620 X-BIN1  206 B=A    X             SAVE EXP IN B[X]
  1059 1621         340 SEL Q                POINTER Q SELECTED
  1060 1622        1334 PT=    13            Q POINTS TO MANTISSA SIGN
  1061 1623         240 SEL P                POINTER P SELECTED
  1062 1624         134 PT=    4             P POINTS TO LOW DIGIT [M]
  1063 1625          12 A=0    WPT           CLEAR A[4:0]
  1064 1626 X-BIN2 1762 A SL   PQ            SHIFT ONE DIGIT INTO A[S]
  1065 1627         116 C=0                  CLEAR ACCUMULATOR REGISTER
  1066 1630         276 AC EX  S             GET THE DIGIT INTO C[S]
  1067 1631        1374 RCR    13            C[0] = DIGIT
  1068 1632        1012 C=A+C  WPT           ADD DIGIT TO PREV NUMBER
  1069 1633         412 A=C    WPT           COPY TO REGISTER A
  1070 1634         146 AB EX  X             GET EXP FROM B[X]
  1071 1635         646 A=A-1  X             DONE WITH CONVERSION ?
  1072 1636        1540 RTN C                YES, RETURN FROM ROUTINE
  1073 1637         146 AB EX  X             PUT REMAINING EXP TO B[X]
  1074 1640         752 C=C+C  WPT           MULTIPLY BY 10 BY DOUBLING
  1075 1641         752 C=C+C  WPT            C[4:0] THREE TIMES AND
  1076 1642         752 C=C+C  WPT            ADDING C[4:0] PREVIOUSLY
  1077 1643        1012 C=A+C  WPT            STORED IN REG A TWICE
  1078 1644         512 A=A+C  WPT           OVERFLOW CONDITION ?
  1079 1645        1613 GONC   X-BIN2 (1626) NO, CONTINUE CONVERSION
  1080 1646 X-BER     1 GOSUB  IFC           SEND INTERFACE CLEAR CMD
  1080 1647           0                      *ILCAS&CTL:  CS3, @0113
  1081 1650           1 GOLONG ERRDE         SHOW MESSAGE: DATA ERROR
  1081 1651           2                      *MAINFRAME: CN10, @0055
*********************************************************************
* RDLPBK - READ  DEVICE BUFFER 1 IN LOOPBACK MODE                   *
* WRLPBK - WRITE DEVICE BUFFER 1 IN LOOPBACK MODE                   *
*********************************************************************
  1086                  ENTRY  RDLPBK
  1087                  ENTRY  WRLPBK
*
  1089 1652 RDLPBK    1 GOSUB  TALKER        ADDRESS DEVICE AS A TALKER
  1089 1653           0                      *ILCAS&CTL:  CS0, @0262
  1090 1654         460 LDI                  DEVICE DEPENDENT TALKER 1
  1091 1655         301 CON    @301          DDT1 (READ BUFFER 1)
  1092 1656 SCMDJ1    1 GOLONG SCMD          SEND COMMAND TO DEVICE
  1092 1657           2                      *ILCAS&CTL:  CS0, @0272
*
  1094 1660 WRLPBK  460 LDI                  DEV DEPENDENT LISTENER 1
  1095 1661         241 CON    @241          DDL1 (WRITE BUFFER 1)
```

```
  1096 1662        1743 GOTO   SCMDJ1 (1656) SEND COMMAND TO DEVICE
*
*************************************************************************
* PLEREX - PIL ERROR EXIT                                               *
*       1. SEND AN "IFC" (INTERFACE CLEAR) COMMAND                      *
*       2. CALL "ERRSUB" BUT WILL NOT RETURN IF ERROR IGNORE FLAG SET   *
*       3. CLEAR LCD DISPLAY                                            *
*       4. CALL "MESSL" SO THE CONSTANT STRING FOLLOWING "PLEREX"       *
*          WILL BE DISPLAYED AS AN ERROR MESSAGE                        *
*************************************************************************
  1106                   ENTRY  PLEREX
*
  1108 1663 PLEREX     1 GOSUB  IFC           SEND INTERFACE CLEAR CMD
  1108 1664            0                      *ILCAS&CTL:  CS3, @0113
  1109 1665            1 GOSUB  ERRSUB        ERROR EXIT SUBROUTINE
  1109 1666            0                      *MAINFRAME:  CN8, @1350
*
*************************************************************************
* MESSLP - PRINTER ROM ENTRY FOR MESSL SUBROUTINE IN MAINFRAME          *
*                                                                       *
* USES:    C[6:0], 1 ADDITIONAL SUBROUTINE LEVEL & LEAVES LCD ENABLED   *
* INPUT:   FOLLOWING THE GOSUB, A SERIES OF CONSTANTS GIVING THE LCD    *
*          FORM OF THE CHARACTERS IN THE MESSAGE, FROM LEFT TO RIGHT.   *
*          LAST CHAR SHOULD HAVE @1000 ADDED.  SPECIAL CHARACTERS       *
*          (THOSE HAVING LCD CREG=1) CAN ONLY BE USED AS THE FINAL      *
*          CHARACTER OF THE MESSAGE.                                    *
* OUTPUT:  LEAVES LCD ENABLED.                                          *
* ASSUMES: HEX MODE                                                     *
*************************************************************************
  1123                   ENTRY  MESSLP
  1124 1667 MESSLP     1 GOSUB  CLLCDE        ENABLE AND CLEAR LCD
  1124 1670            0                      *MAINFRAME: CN11, @0360
  1125 1671            1 GOLONG MESSL         LEFT SHIFT INTO LCD F/RIGHT
  1125 1672            2                      *MAINFRAME:  CN1, @1757
*
*************************************************************************
* PWRUP - GROUP POWER UP                                                *
*************************************************************************
*
  1131                   ENTRY  PWRUP
*
  1133 1673          220 CON    @220          P
  1134 1674           25 CON    @25           U
  1135 1675           22 CON    @22           R
  1136 1676           27 CON    @27           W
  1137 1677           20 CON    @20           P
  1138 1700 PWRUP      1 GOSUB  WKUPLP        WAKE UP HPIL LOOP
  1138 1701            0                      *ILCAS&CTL:  CS1, @1706
  1139 1702            1 GOSUB  PLERCK        CHECK IF LOOP INTACT
  1139 1703            0                      *ILCAS&CTL:  CS1, @1747
  1140 1704            1 GOLONG RSIDY         SET AUTO IDENTIFY BIT
  1140 1705            2                      *ILCAS&CTL:  CS0, @1410
*
  1142                   ENTRY  WKUPLP
*
  1144 1706 WKUPLP  1110 S9=     1            SET ERROR FLAG
  1145 1707         1670 C=REGN  14           READ FLAGS REGISTER
  1146 1710          574 RCR     6            C[S] = FLAGS 32-35
  1147 1711          776 C=C+C   S            MOVE FLAG 32 TO CARRY
  1148 1712          776 C=C+C   S            IN DEVICE MODE (FLAG 33) ?
```

```
  1149 1713         1540 RTN C                 YES, RETURN TO CALLER
  1150 1714          344 HPL=CH 3              WRITE PARALLEL POLL REG
  1151 1715            1 CH=    @000           TURN ON THE PIL CHIP
  1152 1716           44 HPL=CH 0              WRITE STATUS REGISTER
  1153 1717            5 CH=    @001           SET MASTER CLEAR FLAG
  1154 1720           44 HPL=CH 0              WRITE STATUS REGISTER
  1155 1721         1701 CH=    @360           SET SC=CA=TA=LA=1
  1156 1722          144 HPL=CH 1              WRITE CONTROL INT REGISTER
  1157 1723         1405 CH=    @301           IDENTIFY CLASS (BITS 7 & 6)
  1158 1724          460 LDI                   LOAD LOW 12 BITS OF C WITH
  1159 1725           35 CON    29             CODE TO WAKE UP 30 DEVICES
  1160 1726          406 A=C    X              A[2:0] = 29
  1161 1727 WKUP10   460 LDI                   LOAD LOW 12 BITS OF C WITH
  1162 1730           43 CON    35             30 MILLISEC FOR EACH DEV.
  1163 1731         1200 HPIL=C 2              WRITE TO DATA REGISTER
  1164 1732 WKUP20   454 FRAV?                 ANY IDY COMES BACK YET ?
  1165 1733           67 GOC    WKUP30 (1741)  YES, IDENTIFY CAME BACK
  1166 1734         1146 C=C-1  X              TIME OUT FOR 50 MSEC YET ?
  1167 1735         1753 GONC   WKUP20 (1732)  NOT YET, KEEP LOOPING
  1168 1736          646 A=A-1  X              TRY 30 TIMES YET ?
  1169 1737         1703 GONC   WKUP10 (1727)  NOT YET, TRY NEXT DEVICE
  1170 1740           23 GOTO   WKUP40 (1742)  YES, 30 TRIES, EXIT NOW
  1171 1741 WKUP30  1104 S9=    0              CLEAR ERROR FLAG
  1172 1742 WKUP40    44 HPL=CH 0              WRITE STATUS REGISTER
  1173 1743            5 CH=    @001           SET MASTER CLEAR FLAG
  1174 1744         1740 RTN                   END OF POWER UP ROUTINE
*
  1176                   ENTRY  PILERR
  1177                   ENTRY  PLERCK
  1178                   ENTRY  ERRRTN
  1179                   ENTRY  UNTCHK
*
  1181 1745 UNTCHK     1 GOSUB  UNT             SEND UNTALK COMMAND
  1181 1746            0                        *ILCAS&CTL:  CS0, @0254
  1182 1747 PLERCK  1114 ?S9=1                  ERROR FLAG SET ?
  1183 1750         1640 RTN NC                 NO, RETURN TO CALLER
  1184 1751 PILERR     1 GOSUB  PLEREX          ERROR EXIT: TRANSMIT ERR
  1184 1752            0                        *ILCAS&CTL:  CS1, @1663
  1185 1753           24 CON    @24            T
  1186 1754           22 CON    @22            R
  1187 1755            1 CON    @01            A
  1188 1756           16 CON    @16            N
  1189 1757           23 CON    @23            S
  1190 1760           15 CON    @15            M
  1191 1761           11 CON    @11            I
  1192 1762           24 CON    @24            T
  1193 1763           40 CON    @40                 BLANK
  1194 1764            5 CON    @05            E
  1195 1765           22 CON    @22            R
  1196 1766         1022 CON    @1022          R
  1197 1767 ERRRTN     1 GOSUB  LEFTJ           DISPLAY LEFT-JUSTIFIED
  1197 1770            0                        *MAINFRAME: CN10, @1767
  1198 1771            1 GOSUB  ENCP00          ENABLE CHIP 0
  1198 1772            0                        *MAINFRAME:  CN2, @0522
  1199 1773            1 GOSUB  MSGDLY          DELAY FOR VIEWING MESSAGE
  1199 1774            0                        *MAINFRAME:  CN0, @1574
  1200 1775            1 GOLONG ERR110          ERROR EXIT W/WO BACKSTEP
  1200 1776            2                        *MAINFRAME:  CN8, @1373
*
*
```

```
*
  1204                  UNLIST
  1207                  END

  ERRORS :      0
```

```
SYMBOL TABLE (SCPL2B - ILCAS&CTL QUAD 1 = CS1 = ADDRESSES @72000-73777)

B-D10    343   -   345
B-D20    360   -   363
B-D30    364   -   361
B-DEC    330   -   316
B-RG10   323   -   325
BLDAPC   274   -   266  264
BLDAPH   273   -   306
CASSET    13   -
CREATF  1214   -
CRF10   1267   -  1265
CRF20   1334   -  1416 1410 1377
CRF2ND  1277   -  1371
CRF40   1372   -  1347
CRF45   1427   -  1364
CRT10   1241   -  1221
CRTFL   1260   -
CRTFL0  1256   -
CRTFLX  1245   -
CSCKUT   654   -   703
CSEOF   1474   -
CSERCK  1064   -  1057
CSERR     42   -   130
CSERX1  1511   -  1462
CSERX2  1607   -  1511
CSFULL  1437   -
CSSTCK    40   -
DIR       71   -
DIR150   133   -   143
DIR20    131   -
DIRFUL  1446   -  1436
DIRROM   126   -
DSDLY   1466   -
DSERJ     64   -    55
DSFULL  1453   -  1445
DUPFL   1222   -
ERRRTN  1767   -
FILTYP   153   -
FLTP20   225   -   220
FLTP30   233   -   226
FLTP40   241   -   234
FLTPS    221   -
FLTYOP   214   -   210  204  200  174  170  164
LJDLY   1463   -
LOPERR  1072   -  1070
MESSLP  1667   -
NEWTAP  1516   -
NONXT   1435   -  1370
NOTAPE    13   -    45
NWTP10  1527   -  1525
PADBLK   247   -   257
PARWRT    26   -
PILERR  1751   -
PLERCK  1747   -
PLEREX  1663   -
PRBI10   366   -   412
PRBI40   400   -   375
```

```
PRBI45   405   -   377
PRBIN#   307   -   253
PRT11?   424   -
PWRUP   1700   -
RDLPBK  1652   -
RDREG   1020   -
RDREG0  1017   -  1001
RDRG10  1023   -
RDRG15  1025   -  1046
RDRG20  1134   -  1127
RDRG30  1047   -  1043
RDRG40  1062   -  1121
RDRG50  1116   -  1133
RDRGA   1076   -
RDRGDN  1054   -  1051
RDRS10   777   -   765
READR   1007   -
READRX   753   -
RENAME   506   -
RNAM10   532   -   477
SCMDJ1  1656   -  1662
SCMDWT    30   -
SEC      446   -
SEEKR   1144   -
SEQ10    462   -   451
SNDR10   625   -   644
SNDR20   732   -   743
SNDR30   646   -   627
SNDRDN   650   -
SNDRG0   620   -   575
SNDRGA   623   -
SNDRGC   717   -
SNDRGS   621   -
TAPERR    56   -    43
TRYNXT  1366   -  1341
TYPASC   260   -
TYPS10   261   -   272
UNSEC    457   -
UNTCHK  1745   -
WAITS     32   -    37
WKUP10  1727   -  1737
WKUP20  1732   -  1735
WKUP30  1741   -  1733
WKUP40  1742   -  1740
WKUPLP  1706   -
WRERCK   660   -   712  645
WRLPBK  1660   -
WRRS10   575   -   556
WRTR     602   -
WRTRX    542   -
WRTRXX   612   -
X-BER   1646   -
X-BIN   1612   -
X-BIN1  1620   -  1616
X-BIN2  1626   -  1645
X-BINC  1614   -
ZERO     666   -
ZERO10   702   -   716
ZERO20   706   -   714
ZEROFL   675   -
```

ENTRY TABLE (SCPL2B - ILCAS&CTL QUAD 1 = CS1 = ADDRESSES @72000-73777)

```
BLDAPC   274   -
BLDAPH   273   -
CASSET    13   -
CREATF  1214   -
CRF2ND  1277   -
CRTFL   1260   -
CRTFL0  1256   -
CRTFLX  1245   -
CSCKUT   654   -
CSEOF   1474   -
CSERCK  1064   -
CSERR     42   -
CSSTCK    40   -
DIR       71   -
DIR150   133   -
DIRROM   126   -
DSDLY   1466   -
DUPFL   1222   -
ERRRTN  1767   -
LJDLY   1463   -
MESSLP  1667   -
NEWTAP  1516   -
PARWRT    26   -
PILERR  1751   -
PLERCK  1747   -
PLEREX  1663   -
PRT11?   424   -
PWRUP   1700   -
RDLPBK  1652   -
RDREG   1020   -
RDREG0  1017   -
RDRG10  1023   -
RDRGA   1076   -
READR   1007   -
READRX   753   -
RENAME   506   -
RNAM10   532   -
SCMDWT    30   -
SEC      446   -
SEEKR   1144   -
SNDR10   625   -
SNDRDN   650   -
SNDRG0   620   -
SNDRGA   623   -
SNDRGC   717   -
SNDRGS   621   -
TAPERR    56   -
TYPASC   260   -
UNSEC    457   -
UNTCHK  1745   -
WAITS     32   -
WKUPLP  1706   -
WRLPBK  1660   -
WRTR     602   -
WRTRX    542   -
WRTRXX   612   -
```

X-BER	1646	-
X-BIN	1612	-
X-BINC	1614	-
ZERO	666	-
ZEROFL	675	-

EXTERNAL REFERENCES (SCPL2B - ILCAS&CTL QUAD 1 = CS1 = ADR @72000-73777)

ACKX	1612					
ACKX	1613					
AOUT1	507					
AOUT1	510					
AOUTFL	526	530	1427			
AOUTFL	527	531	1430			
APPEND	302	304	405			
APPEND	303	305	406			
ARGOUT	243					
ARGOUT	244					
BINBD0	332					
BINBD0	333					
BLDAPH	75	150	221	227	235	254
BLDAPH	76	151	222	230	236	255
CHKCS0	1537					
CHKCS0	1540					
CHKCST	71	542	753			
CHKCST	72	543	754			
CHKPCT	522	550	604	670		
CHKPCT	523	551	605	671		
CLA	73	144				
CLA	74	145				
CLLCDE	1667					
CLLCDE	1670					
CRTFLX	1241					
CRTFLX	1242					
CSEOF	1163					
CSEOF	1164					
CSERCK	660					
CSERCK	661					
CSEREX	24	1237	1607			
CSEREX	25	1240	1610			
CSERR	1074					
CSERR	1075					
CSSTAS	32	1065				
CSSTAS	33	1066				
DATALL	610	1013				
DATALL	611	1014				
DFBDCK	552	761				
DFBDCK	553	762				
DIR150	422					
DIR150	423					
DSDLY	245					
DSDLY	246					
DSPERR	64					
DSPERR	65					
DTFLOW	621					
DTFLOW	622					
DUPFL	516					
DUPFL	517					
ENCP00	1771					
ENCP00	1772					
ERR110	1775					
ERR110	1776					
ERRDE	1533	1650				
ERRDE	1534	1651				

```
UNT      137   656  1060  1745
UNT      140   657  1061  1746
WAITS    654  1601
WAITS    655  1602
WKUPLP  1700
WKUPLP  1701
X-BER   1250
X-BER   1251
X-BIN   1150  1245
X-BIN   1151  1246
ZEROFL  1243  1553
ZEROFL  1244  1554

End of VASM assembly
********************************************************************************
********************************************************************************
********************************************************************************

VASM ROM ASSEMBLY          REV.  6/81A            HP-82160A HP-IL MODULE

OPTIONS: L C S          HP-IL CAS&CTL             ADDRESSES @74000-75777

     2                  FILE   SCPL3B             ILCAS&CTL QUAD 2 = CS2
********************************************************************************
* FLSCH  - SEARCH FOR FILE.  WILL GENERATE ERROR MESSAGE IF THE NAMED  *
*          FILE IS FOUND (C[S] = 0) OR IF IT IS NOT FOUND (C[S] > 0).  *
*                                                                      *
* INPUT :                                                              *
* C[S]=  0 SEARCH FOR DUPLICATE FILE NAME,      ERROR IF FOUND         *
* C[S]=  8 SEARCH FOR PROGRAM FILE,             ERROR IF NOT FOUND     *
* C[S]= 13 SEARCH FOR DATA FILE,                ERROR IF NOT FOUND     *
* C[S]=  4 SEARCH FOR WRITE ALL FILE,           ERROR IF NOT FOUND     *
* C[S]=  5 SEARCH FOR KEY FILE,                 ERROR IF NOT FOUND     *
* C[S]= 14 SEARCH A FILE, DON'T CARE WHAT TYPE, ERROR IF NOT FOUND     *
* C[S]= 15 SAME AS 14 EXCEPT SEARCHING ALWAYS STARTS FROM RECORD 0     *
*          EXCEPT FOR C[S]=15, ALL THE FILE SEARCHING STARTS FROM      *
*          THE DIRECTORY BUFFER                                        *
* S11:     S11 IS USED AS A SPECIAL CASE FOR REWRITE PROGRAM.  IT IS   *
*          SET BY THE REGULAR ENTRY. IF C[S] = 0 AND S11 = 0, IT WILL  *
*          NOT TREAT THE DUPLICATE FILE AS AN ERROR.                   *
*                                                                      *
* USES:    A, B, C, M, N, S[9:0], +3 SUBROUTINE LEVELS                 *
* CAUTION: SINCE FLSCH USES +3 SUBROUTINE LEVELS, IT WILL LOAD THE     *
*          ADDRESS OF NFRPU INTO RETURN STACK BEFORE RETURNING         *
* OUTPUT:  IF THE FILE FOUND:                                          *
*          M = FILE NAME                                               *
*          N = FILE ENTRY                                              *
*          IF FILE NOT FOUND, WILL RETURN WITH M = 0                   *
*          LEAVES CASSETTE NOT A TALKER NOR A LISTENER                 *
********************************************************************************
    30                  ENTRY  FLSCHJ
    31                  ENTRY  FLSCHI
    32                  ENTRY  FLSCH
    33                  ENTRY  FLSCHD
    34                  ENTRY  FLSCHX
    35                  ENTRY  FLTYER
    36                  ENTRY  FLSCHC
    37                  ENTRY  FLSCH0
*
    39    0 NOCST     1 GOLONG CSNOFD        CASSETTE NOT FOUND ROUTINE
```

```
    39    1           2                        *ILCAS&CTL:  CS3, @0364
*
    41    2 FLSCHI  460 LDI                      SEARCH FOR ANY TYPE FILE
    42    3          16 CON     14               14 = DON'T CARE WHAT TYPE
    43    4          63 GOTO    FLSCH0 (  12)    START SEARCHING FILE
    44    5 FLSCHJ  460 LDI                      SEARCH FOR ANY TYPE FILE
    45    6          17 CON     15               15 = START AT RECORD ZERO
    46    7          33 GOTO    FLSCH0 (  12)    START SEARCHING FILE
*
    48   10 FLSCHD  460 LDI                      SEARCH FOR DATA TYPE FILE
    49   11          15 CON     13               13 = DATA FILE TYPE
    50   12 FLSCH0 1474 RCR     1                C[S] = FILE TYPE
    51   13 FLSCH   376 BC EX   S                B[S] = FILE TYPE
    52   14           1 GOSUB   AOUTIN           SET STARTING ADDRESS
    52   15           0                          *ILCAS&CTL:  CS0, @1266
    53   16 FLSCHX    1 GOSUB   FNDCAS           LOOK FOR CASSETTE
    53   17           0                          *ILCAS&CTL:  CS0, @0566
    54   20        1603 GOTO    NOCST  (   0)    P+1 - CASSETTE NOT FOUND
    55   21           1 GOSUB   R5-R6            P+2 - SAVE LO0P ADDR IN R6
    55   22           0                          *ILCAS&CTL:  CS0, @1742
    56   23 FLSCHC  404 S8=     0                FILE SEARCH WITH COPYBF
    57   24         336 C=B     S                C[S] = FILE TYPE TO SEARCH
    58   25        1376 ? C#0   S                NEED TO CHECK NEW TAPE ?
    59   26          33 GONC    FLSH01 (  31)    NO, START FROM REC.2 ANYWAY
    60   27        1076 C=C+1   S                YES, IS THIS FILE TYPE 15 ?
    61   30          23 GONC    FLSH05 (  32)    NO, SOME OTHER FILE TYPE
    62   31 FLSH01  410 S8=     1                YES, DON'T DO COPYBF
    63   32 FLSH05    1 GOSUB   CSRDY            SEE IF CASSETTE IS BUSY
    63   33           0                          *ILCAS&CTL:  CS3, @0341
    64   34        1704 CLR ST                   LEFT SHIFT FILE NAME TO M
    65   35           1 GOSUB   AOUTFL           GET FILE NAME FROM A-REG
    65   36           0                          *ILCAS&CTL:  CS0, @1066
    66   37         630 C=M                      COPY NAME INTO C-REGISTER
    67   40        1150 REGN=C 9                 SAVE FILE NAME IN REG.9
    68   41         336 C=B     S                IF B[S]=15,0 START F/REC 0
    69   42        1376 ? C#0   S                IS SEARCH TYPE 0 ?
    70   43         333 GONC    FLSH20 (  76)    YES, START FROM RECORD 0
    71   44        1076 C=C+1   S                IS SEARCH TYPE 15 ?
    72   45         317 GOC     FLSH20 (  76)    YES, START FROM RECORD 0
    73   46           1 GOSUB   SETBP0           PREPARE TO SEARCH DIR BUFF
    73   47           0                          *ILCAS&CTL:  CS3, @1626
    74   50           1 GOSUB   RDLPBK           GO INTO READ LOOP BACK MODE
    74   51           0                          *ILCAS&CTL:  CS1, @1652
    75   52           6 A=0     X                CLEAR CURRENT ENTRY NUMBER
    76   53 FLSH10  206 B=A     X                STORE CURRENT ENTRY NUMBER
    77   54           1 GOSUB   RENT10           READ NEXT FILE ENTRY
    77   55           0                          *ILCAS&CTL:  CS2, @0243
    78   56         260 C=N                      FILE INFORMATION TO REG C
    79   57        1376 ? C#0   S                IS THE FILE PURGED ?
    80   60         103 GONC    FLSH15 (  70)    YES, GET CURRENT ENTRY #
    81   61        1170 C=REGN 9                 GET TARGET FILE NAME
    82   62         416 A=C                      SAVE TARGET NAME IN REG A
    83   63         630 C=M                      FILE NAME WE JUST READ IN
    84   64        1556 ? A#C                    IS THIS THE ONE ?
    85   65         603 GONC    FLTYCK ( 145)    YES, WE FOUND IT, CHK TYPE
    86   66        1056 C=C+1                    NO, REACHED END OF DIR ?
    87   67          77 GOC     FLSH20 (  76)    YES, LET'S START IT OVER
    88   70 FLSH15  146 AB EX   X                GET CURRENT ENTRY NUMBER
    89   71         546 A=A+1   X                INCREMENT CURRENT ENTRY #
    90   72         460 LDI                      CHECK AGAINST MAX ENTRY #
```

```
 91   73          10 CON    8              LAST ENTRY # IN BUFFER
 92   74        1546 ? A#C  X              THROUGH DIR BUFFER YET ?
 93   75        1567 GOC    FLSH10 (  53) NOT YET, READ NEXT ENTRY
 94   76 FLSH20    1 GOSUB  SEEKR2         SEEK TO RECORD 2 & READ IT
 94   77           0                       *ILCAS&CTL:  CS3, @1557
 95  100        1204 S7=    0              RESET FLAG FOR SEARCH BUFR
 96  101 FLSH30    1 GOSUB  RENTPH         READ ONE DIRECTORY ENTRY
 96  102           0                       *ILCAS&CTL:  CS0, @1770
 97  103 FLSH32  260 C=N                   FILE INFORMATION INTO C
 98  104        1376 ? C#0  S              PURGED FILE ?
 99  105        1743 GONC   FLSH30 ( 101) YES, GET NEXT DIR ENTRY
100  106        1170 C=REGN 9              NO, GET TARGET FILE NAME
101  107         416 A=C                   SAVE TARGET NAME IN REG A
102  110         630 C=M                   FILE NAME WE JUST READ IN
103  111        1556 ? A#C                 IS THIS THE FILE ?
104  112         333 GONC   FLTYCK ( 145) YES, WE FOUND IT, CHK TYPE
105  113        1056 C=C+1                 IS IT END OF DIRECTORY ?
106  114        1653 GONC   FLSH30 ( 101) NOT YET, READ NEXT ENTRY
107  115           1 GOSUB  FNTDEV         LOOK FOR ANOTHER DRIVE
107  116           0                       *ILCAS&CTL:  CS2, @1657
108  117          23 GOTO   FLSH35 ( 121) P+1 - NO OTHER DRIVE FOUND
109  120        1033 GOTO   FLSCHC (  23) P+2 - EXAMINE NEXT DRIVE
110  121 FLSH35  116 C=0                   CLEAR ACCUMULATOR
111  122         530 M=C                   CLEAR REGISTER M
112  123        1336 ? B#0  S              LOOKING FOR DUPLICATES ?
113  124         473 GONC   FLSHRT ( 173) YES, NOT FOUND IS OK
114  125           1 GOSUB  PLEREX         DISPLAY "FL NOT FOUND"
114  126           0                       *ILCAS&CTL:  CS1, @1663
115  127           6 CON    @06            F
116  130          14 CON    @14            L
117  131          40 CON    @40                 BLANK
118  132          16 CON    @16            N
119  133          17 CON    @17            O
120  134          24 CON    @24            T
121  135          40 CON    @40                 BLANK
122  136           6 CON    @06            F
123  137          17 CON    @17            O
124  140          25 CON    @25            U
125  141          16 CON    @16            N
126  142        1004 CON    @1004          D
127  143 FLSHER    1 GOLONG CSEREX         CASSETTE ERROR EXIT
127  144           2                       *ILCAS&CTL:  CS3, @0376
128  145 FLTYCK  336 C=B    S              SEE IF ANY TYPE ACCEPTABLE
129  146        1376 ? C#0  S              B[S] = 0 ?
130  147         243 GONC   FLSHRT ( 173) YES, LOOK FOR DUPLIC. FILE
131  150        1076 C=C+1  S              B[S] = 15 ?
132  151         227 GOC    FLSHRT ( 173) YES, ANY TYPE IS OK
133  152        1076 C=C+1  S              B[S] = 14 ?
134  153          57 GOC    FLSH70 ( 160) YES, DON'T CARE TYPE EITHER
135  154         176 AB EX  S              VERIFY THE FILE TYPE
136  155         260 C=N                   GET FILE INFORMATION FROM N
137  156        1576 ? A#C  S              DO FILE TYPES MATCH ?
138  157         337 GOC    FLTYER ( 212) NO, FILE TYPE ERROR
139  160 FLSH70 1214 ?S7=1                 FILE FOUND IN BUFFER ?
140  161         127 GOC    FLSHRT ( 173) YES, DON'T COPY DIR TO BUFR
141  162           1 GOSUB  REQADR         SEE IF AT LAST ENTRY OF REC
141  163           0                       *ILCAS&CTL:  CS3, @1100
142  164         646 A=A-1  X              BACK UP ONE RECORD
143  165         646 A=A-1  X              BACK UP TWO RECORDS
144  166        1312 ? B#0  WPT            BYTE POINTER = 0 ?
```

```
   145  167           1 GSUBNC SEEKRD        YES, BACK UP 2 REC & READ
   145  170           0                      *ILCAS&CTL:  CS3, @1572
   146  171           1 GOSUB  COPYBF        COPY CURR DIR REC TO BUFFER
   146  172           0                      *ILCAS&CTL:  CS3, @0306
   147                  ENTRY  FLSHRT
   148  173 FLSHRT   660 C=STK                POP LAST ADDR FROM RTN STK
   149  174          432 A=C    M             A[M] = LAST RETURN ADDRESS
   150  175          116 C=0                  CLEAR ACCUMULATOR
   151  176         1176 C=C-1  S             C[S] = F
   152  177         1174 RCR    9             C[6:3] = 00F0 HEX = @0360
   153  200          560 STK=C                PUSH NFRPU ADDR TO STACK
   154  201          272 AC EX  M             GET SAVED RETURN ADDRESS
   155  202          740 GOTOC                GO DIRECTLY TO THAT ADDR
*
   157                  ENTRY  FLSHDT
*
   159  203 FLSHDT   460 LDI                  LOAD LOW 12 BITS OF C WITH
   160  204           15 CON    13            C[1:0] = 13 (DATA FILE)
   161  205         1474 RCR    1             C[S] = 13
   162  206          436 A=C    S             A[S] = 13
   163  207          260 C=N                  C[S] = FILE TYPE
   164  210         1576 ? A#C  S             IS THIS A DATA FILE ?
   165  211         1640 RTN NC               YES, BOTH ARE CODE 13
   166  212 FLTYER     1 GOSUB  PLEREX        DISPLAY "FL TYPE ERR"
   166  213           0                      *ILCAS&CTL:  CS1, @1663
   167  214           6 CON    @06           F
   168  215          14 CON    @14           L
   169  216          40 CON    @40                BLANK
   170  217          24 CON    @24           T
   171  220          31 CON    @31           Y
   172  221          20 CON    @20           P
   173  222        1005 CON    @1005         E
   174  223           1 GOLONG DSPERR        DISPLAY MESSAGE: ERR
   174  224           2                      *ILCAS&CTL:  CS3, @0405
*
*************************************************************************
* RWCHK  - OVERWRITE CHECK                                              *
*          IF A DUPLICATE FILE NAME IS NOT FOUND, RETURN IMMEDIATELY    *
*          IF A DUPLICATE FILE NAME IS FOUND AND FILE TYPE IS RIGHT,    *
*          AND THE FILE IS NOT PROTECTED, WILL PURGE THE FILE.          *
* INPUT:   C[X] = FILE TYPE                                             *
*          IF M = 0, THIS MEANS NO DUPLICATE FILE NAME WAS FOUND        *
*          OTHERWISE, M = DUPLICATE FILE NAME, N = FILE INFORMATION     *
*************************************************************************
   185                  ENTRY  RWCHK
*
   187  225 RWCHK   1474 RCR    1             C[S] = FILE TYPE
   188  226          436 A=C    S             A[S] = FILE TYPE
   189  227          630 C=M                  0 NO DUP, OTHERWISE FLNAME
   190  230         1356 ? C#0                FOUND DUPLICATE FILE NAME ?
   191  231         1640 RTN NC               NO, RETURN IMMEDIATELY
   192  232 RWCK20   260 C=N                  CHECK IF THE SAME TYPE
   193  233         1576 ? A#C  S             ARE BOTH FILES SAME TYPE ?
   194  234           1 GOLC   DUPFL          NO, DISPLAY "DUP FL NAME"
   194  235           3                      *ILCAS&CTL:  CS1, @1222
   195  236         236 B=A    S             GET TYPE BACK TO B[S]
   196  237           1 GOLONG PURGEP         CHECK & PURGE IF NOT PROT
   196  240           2                      *ILCAS&CTL:  CS3, @0140
   197                  FILLTO @240
*************************************************************************
```

```
* RDENT  - READ FILE ENTRY                                                *
* ASSUMES: CASSETTE IS AN ADDRESSED TALKER                                *
* USES:    A, C, S6, +2 SUBROUTINE LEVELS                                 *
* OUTPUT:  M = FILE NAME IN ASCII                                         *
*          N = FILE INFORMATION                                           *
*          N[13:12] = FILE TYPE                                           *
*                     N[13] = 1 - ASCII DATA FILE,    S6=0                *
*                             4 - 41C WRITE ALL FILE, S6=0                *
*                             5 - 41C KEY FILE,       S6=0                *
*                             6 - 41C STATUS FILE,    S6=0                *
*                             8 - 41C PROGRAM FILE,   S6=0                *
*                            13 - FIXED LENGTH                            *
*                                 (8-BYTE) DATA FILE, S6=0                *
*                            15 - NOT A 41C FILE,     S6=1                *
*                     N[12] = FILE PROTECTION INFORMATION                 *
*                     N[12] = -----------------                           *
*                             | S | 0 | A | P |                           *
*                             -----------------                           *
*          N[11:8]  = FILE STARTING RECORD NUMBER                         *
*          N[7:4]   = FILE LENGTH IN NUMBER OF RECORDS                    *
*          N[3:0]   = FILE SIZE IN BYTES (PROG FILE) OR IN REGISTERS      *
* RETURNS: CASSETTE AS A TALKER                                           *
*************************************************************************
*
   223                  ENTRY  RDENT
   224                  ENTRY  RENT10
*
   226  241 RDENT     1 GOSUB  DDT0          SEND SEC.CMD - SEND DATA
   226  242            0                     *ILCAS&CTL:  CS0, @0320
   227  243 RENT10     1 GOSUB  NATNRD        SEND SDA, WAIT FOR A BYTE
   227  244            0                     *ILCAS&CTL:  CS0, @0352
   228  245          434 PT=    8             SET UP TO READ 8 BYTES
   229  246 RENT15     1 GOSUB  RDDFRM        READ 7 BYTES OF NAME
   229  247            0                     *ILCAS&CTL:  CS0, @0420
   230  250         1200 HPIL=C 2             RETRANSMIT LAST DATA FRAME
   231  251            1 GOSUB  PLERCK        CHECK FOR ANY ERRORS
   231  252            0                     *ILCAS&CTL:  CS1, @1747
   232  253         1724 DEC PT               MOVE TO NEXT BYTE
   233  254         1624 ? PT=  0             READ 8 BYTES YET ?
   234  255           47 GOC    RENT20 ( 261) YES, GET FILE TYPE NEXT
   235  256            1 GOSUB  CX-AX         NO, SAVE THE BYTE IN REG A
   235  257            0                     *ILCAS&CTL:  CS2, @0362
   236  260         1663 GOTO   RENT15 ( 246) READ NEXT BYTE OF NAME
   237  261 RENT20   256 AC EX                GET FILENAME FROM C TO A
   238  262          530 M=C                  SAVE FILENAME IN REG M
   239  263 RENT30  1034 PT=    2             SKIP NEXT 2 BYTES AND
   240  264           16 A=0                    READ 1ST BYTE OF TYPE
   241  265            1 GOSUB  SKPFRM        READ AND DISCARD FRAMES
   241  266            0                     *ILCAS&CTL:  CS2, @0341
   242  267         1166 C=C-1  XS            C[XS] = F
   243  270         1046 C=C+1  X             TEST IF TYPE = -1 ?
   244  271           33 GONC   RENT40 ( 274) NO, SOME OTHER TYPE
   245  272         1156 C=C-1                YES, INDICATE END OF DIR
   246  273          530 M=C                  M = FFFF.... (END OF DIR)
   247  274 RENT40     1 GOSUB  SKPFRM        READ 2ND BYTE OF TYPE
   247  275            0                     *ILCAS&CTL:  CS2, @0341
   248  276          460 LDI                  LOAD LOW 12 BITS OF C WITH
   249  277            1 CON    1             C[1:0] = 1 (ASCII FILE)
   250  300         1546 ? A#C  X             IS FILE TYPE = 1 ?
   251  301           37 GOC    RENT45 ( 304) NO, FILE TYPE IS NOT 1
```

```
   252  302         1756 A SL                    SHIFT TYPE TO RIGHT PLACE
   253  303           63 GOTO   RENT50 ( 311) GET MORE FILE INFORMATION
   254  304 RENT45 1502 ? A#0  PT              RIGHT TYPE FOR 41C ?
   255  305           43 GONC   RENT50 ( 311) YES, GET MORE FILE INFO
   256  306          460 LDI                   UNRECOGNIZED TYPE
   257  307          360 CON2   15     0        MAKE IT A TYPE "F"
   258  310          406 A=C    X              A[X] = FILE TYPE & PROTN
   259  311 RENT50 1034 PT=    2              SKIP 2 BYTES AND READ
   260  312            1 GOSUB  SKPFRM          1ST BYTE OF START REC #
   260  313            0                       *ILCAS&CTL:  CS2, @0341
   261  314            1 GOSUB  SKPFRM         READ 2ND BYTE OF START #
   261  315            0                       *ILCAS&CTL:  CS2, @0341
   262  316         1034 PT=    2              SKIP 2 BYTES AND READ 1ST
   263  317            1 GOSUB  SKPFRM          BYTE OF FILE LENGTH
   263  320            0                       *ILCAS&CTL:  CS2, @0341
   264  321            1 GOSUB  SKPFRM         READ 2ND BYTE OF FILE LEN
   264  322            0                       *ILCAS&CTL:  CS2, @0341
   265  323          434 PT=    8              SKIP 8 BYTES AND READ 1ST
   266  324            1 GOSUB  SKPFRM          BYTE OF FILE SIZE
   266  325            0                       *ILCAS&CTL:  CS2, @0341
   267  326            1 GOSUB  SKPFRM         READ 2ND BYTE OF FILE SIZE
   267  327            0                       *ILCAS&CTL:  CS2, @0341
   268  330            1 GOSUB  RDDFRM         READ 2ND BYTE OF TYPE
   268  331            0                       *ILCAS&CTL:  CS0, @0420
   269  332         1200 HPIL=C 2              ECHO BYTE READ TO LOOP
   270  333         1074 RCR    2              ROTATE DATA 1 BYTE RIGHT
   271  334          136 C=0    S              C=0X000...(C[12]=FL  OPTN)
   272  335         1560 C=CORA                COMBINE BITS FROM A AND C
   273  336          160 N=C                   SAVE FILE INFO TO REG N
   274  337            1 GOLONG NRD            READ LAST BYTE OF ENTRY
   274  340            2                       *ILCAS&CTL:  CS0, @0364
   275                   FILLTO @340
*******************************************************************************
* SKPFRM - READ A GIVEN NUMBER OF FRAMES WITHOUT STORING THEM                 *
*          SKPFRM FALLS INTO CX-AX, SO LAST BYTE IS SAVED IN A[1:0]           *
* CX-AX  - SHIFT A TO LEFT BY ONE BYTE AND SAVE THE BYTE IN C[1:0]            *
*          TO A[1:0]                                                          *
*******************************************************************************
   282                   ENTRY  SKPFRM
   283                   ENTRY  CX-AX
*
   285  341 SKPFRM 1114 ?S9=1                  ERROR FLAG SET ?
   286  342         1540 RTN C                 YES, RETURN IMMEDIATELY
   287  343          106 C=0    X              NO, CLEAR NUMBER OF TRIES
   288  344 SKPF10  454 FRAV?                  DATA FRAME AVAILABLE ?
   289  345           57 GOC    SKPF30 ( 352) PROCESS THE DATA
   290  346         1046 C=C+1  X              EXCEEDED MAXIMUM TRIES ?
   291  347           27 GOC    SKPF20 ( 351) YES, SET ERROR FLAG
   292  350         1743 GOTO   SKPF10 ( 344) NO, LOOK FOR FRAME AGAIN
   293  351 SKPF20 1110 S9=    1              SET ERROR FLAG
   294  352 SKPF30  244 C=HPIL 2              READ DATA REGISTER
   294  353          272                       (INSERTED BY ASSEMBLER)
   294  354          203                       (INSERTED BY ASSEMBLER)
   295  355         1200 HPIL=C 2              ECHO BYTE JUST READ
   296  356         1624 ? PT=  0              READ ALL REQUESTED BYTES ?
   297  357           37 GOC    CX-AX  ( 362) SAVE THE FINAL BYTE READ
   298  360         1724 DEC PT                DECREMENT BYTE COUNTER
   299                   LEGAL                 (CLEAR THE CARRY FLAG)
   300  361         1603 GOTO   SKPFRM ( 341) GET NEXT BYTE
   301  362 CX-AX  1756 A SL                   SHIFT A LEFT BY 1 BYTE
```

```
   302  363        1756 A SL                (1 DIGIT PER EACH SHIFT)
   303  364         266 AC EX  XS           COPY EXPONENT SIGN INTO C
   304  365         406 A=C    X            COPY FINAL BYTE INTO A[X]
   305  366        1740 RTN                 END OF SKPFRM & CX-AX
*
*****************************************************************
* WRTP  -  WRITE CURRENT PROGRAM AS A REGULAR PROGRAM           *
* WRTPV -  WRITE CURRENT PROGRAM AS A PRIVATE PROGRAM           *
*                                                               *
* INPUT:   ALPHA REG - FILE NAME                                *
*          BOTH FUNCTIONS ARE NON-PROGRAMMABLE, AND REQUIRE THAT A
*          PROGRAM NAME BE KEYED IN BEFORE THE PROGRAM IS RECORDED.
*          TRYING TO WRITE A PROGRAM IN ROM OR TO WRITE A PRIVATE
*          PROGRAM IS NOT ALLOWED.                              *
*****************************************************************
*
   318                 ENTRY  WRTP
   319                 ENTRY  WRTPV
   320                 ENTRY  WPROM
*
   322  367         226 CON    @226          V
   323  370          20 CON    @20           P
   324  371          24 CON    @24           T
   325  372          22 CON    @22           R
   326  373          27 CON    @27           W
   327  374 WRTPV   610 S11=   1             SET PRIVATE FLAG
   328  375          63 GOTO   WRTP00 ( 403) GET NAME INTO M
   329  376         220 CON    @220          P
   330  377          24 CON    @24           T
   331  400          22 CON    @22           R
   332  401          27 CON    @27           W
   333  402 WRTP    604 S11=   0             RESET PRIVATE FLAG
   334  403 WRTP00   10 S3=    1             GET NAME TO M SHIFT F/LEFT
   335  404         116 C=0                  CLEAR ACCUMULATOR
   336  405         530 M=C                  CLEAR REGISTER M
   337  406        1010 S2=    1             SAVE LAST CHAR ADDR IN R.8
   338  407           1 GOSUB  AOUT1         GET FILE NAME INTO REG M
   338  410           0                      *ILCAS&CTL:  CS0, @1064
   339  411         630 C=M                  FILE NAME TO REGISTER C
   340  412        1150 REGN=C 9             SAVE FILE NAME TO REG.9
   341  413        1356 ? C#0                LABEL PRESENT ?
   342  414         143 GONC   WRTP10 ( 430) NO, WRITE CURRENT PROGRAM
   343  415           1 GOSUB  ASRCH         YES, SEARCH THE PROGRAM
   343  416           0                      *MAINFRAME:  CN9, @1305
   344  417        1356 ? C#0                PROGRAM FOUND ?
   345  420 WPNFER    1 GOLNC  FLNMER        NO, DISPLAY "NAME ERR"
   345  421           2                      *ILCAS&CTL:  CS0, @1256
   346  422         416 A=C                  YES, COPY FILE NAME TO A
   347  423        1114 ?S9=1                MICROCODE PROGRAM ?
   348  424        1747 GOC    WPNFER ( 420) YES, DISPLAY "NAME ERR"
   349  425        1014 ?S2=1                NO, IS PROGRAM IN ROM ?
   350  426          47 GOC    WPROM  ( 423) YES, DON'T RECORD IT
   351  427         103 GOTO   WRTP20 ( 437) NO, CONTINUE WRITING
   352  430 WRTP10  314 ?S10=1               ROM ?
   353  431          43 GONC   WRTP15 ( 435) NO, GET PROGRAM COUNTER
   354  432 WPROM     1 GOSUB  ERROR         EXIT WITH ERROR MESSAGE
   354  433           0                      *MAINFRAME:  CN8, @1365
   355  434           0 XDEF   MSGROM        ERROR MESSAGE: ROM
   356  435 WRTP15    1 GOSUB  GETPC         GET PROGRAM COUNTER
   356  436           0                      *MAINFRAME: CN10, @0520
```

```
357  437 WRTP20  256 AC EX               COPY PRGM ADDR TO C[3:0]
358  440         374 RCR    10           ROTATE PRGM ADDR TO C[7:4]
359  441         372 BC EX  M            SAVE PRGM ADDR IN B[7:4]
360  442         214 ?S5=1               FILE NAME FOLLOWS ?
361  443           1 GSUBNC AOUTIN       NO, USE PGM NAM AS FL NAM
361  444           0                     *ILCAS&CTL:  CS0, @1266
362  445          76 B=0    S            SET FILE CODE 0
363  446           1 GOSUB  FLSCHX       SEARCH FOR DUPLICATE FILE
363  447           0                     *ILCAS&CTL:  CS2, @0016
364  450         460 LDI                 LOAD LOW 12 BITS OF C WITH
365  451          10 CON    8            TYPE 8 IS A PROGRAM FILE
366  452           1 GOSUB  RWCHK        CHECK DUPLICATE FILE
366  453           0                     *ILCAS&CTL:  CS2, @0225
367  454          34 PT=    3            POINT AT LOWEST 2 BYTES
368  455         316 C=B                 PROGRAM ADDRESS TO C
369  456         174 RCR    4            C[3:0] = PROGRAM ADDRESS
370  457         252 AC EX  WPT          A[3:0] = PROGRAM ADDRESS
371  460           1 GOSUB  FLINKA       FIND PROGRAM END ADDRESS
371  461           0                     *MAINFRAME: CN10, @0447
372  462        1514 ?S12=1              PRIVATE ?
373  463           1 GOLC   ERRPR        YES, ERROR: PRIVATE
373  464           3                     *MAINFRAME:  CN8, @0604
374  465         416 A=C                 A[11:8] = PRGM END ADDRESS
375  466         474 RCR    8            C[3:0] = PRGM END ADDRESS
376  467         412 A=C    WPT          A[3:0[ = PRGM END ADDRESS
377  470           1 GOSUB  CPGMHD       FIND PROGRAM HEAD
377  471           0                     *MAINFRAME:  CN1, @1173
378  472           1 GOSUB  LDSST0       ENABLE CHIP 0, LOAD REG 14
378  473           0                     *MAINFRAME:  CN1, @1627
379  474         674 RCR    11           C[1:0] = FLAGS 4-11
380  475        1530 ST=C               S0 = USER FLAG 11
381  476         256 C=A                 C[11:8]=END, C[3:0]=HEAD
381  477         416                     (INSERTED BY ASSEMBLER)
382  500         374 RCR    10           C[7:4] = PRGM HEAD ADDRESS
383  501         372 B=C    M            B[7:4] = PRGM HEAD ADDRESS
383  502         332                     (INSERTED BY ASSEMBLER)
384  503         212 B=A    WPT          B[3:0] = PRGM HEAD ADDRESS
385  504        1574 RCR    12           C[3:0] = PRGM END ADDRESS
386  505         412 A=C    WPT          A[3:0] = PRGM END ADDRESS
387  506           1 GOSUB  INCAD2       POINT TO 3RD BYTE OF END
387  507           0                     *MAINFRAME: CN10, @0723
388  510         152 AB EX  WPT          B[3:0] = PRGM END ADDRESS
389  511           1 GOSUB  CNTBYT       COUNT PRGM LENGTH IN BYTES
389  512           0                     *ILCAS&CTL:  CS3, @1647
390  513         206 B=A    X            B[X]=FILE SIZE IN # BYTES
391  514         316 C=B                 C[7:4] = PROG HEAD ADDRESS
392  515         374 RCR    10           REG.9[11:8]=PROG HEAD ADDR
393  516        1150 REGN=C 9            REG.9[6:4]=FILE SIZE BYTES
394  517         174 RCR    4            C[X] = FILE SIZE IN BYTES
395  520         132 C=0    M            CLEAR ACCUMULATOR MANTISSA
396  521        1046 C=C+1  X            INCREMENT FILE SIZE BYTES
397  522         374 RCR    10           C[8:4]=FILE LENGTH BYTES
398  523         246 AC EX  X            BRING FILE SIZE BYTES TO C
399  524         102 C=0    PT           C[3:0]=FILE SIZE IN BYTES
400  525        1574 RCR    12           C[5:2]=FILE SIZE IN BYTES
401  526        1434 PT=    1            POINT AT LOWEST BYTE
402  527        1020 LC     8            C[1] = FILE TYPE (PROGRAM)
403  530        1614 ?S0=1               SHOULD PRGM BE AUTO RUN ?
404  531          33 GONC   WRTP30 ( 534) NO, CHECK FOR PRIVATE PRGM
405  532        1042 C=C+1  PT           YES, SET AUTO RUN BIT
```

```
   406  533         1042 C=C+1  PT             C[0] = 2
   407  534 WRTP30   614 ?S11=1                WRITE PRIVATE PROGRAM ?
   408  535           23 GONC   WRTP40 ( 537) NO, GO CREATE THE FILE
   409  536         1042 C=C+1  PT             YES, SET PRIVATE BIT
   410                    LEGAL                (CLEAR THE CARRY FLAG)
   411  537 WRTP40     1 GOSUB  CRTFL          CREATE THE PROGRAM FILE
   411  540            0                       *ILCAS&CTL:  CS1, @1260
   412  541            1 GOSUB  SEKSUB         SEEK & SEND SERIES WRITE
   412  542            0                       *ILCAS&CTL:  CS0, @0326
   413  543          144 HPL=CH 1              WRITE CONTROL INT REGISTER
   414  544            5 CH=    @001           ENABLE FI LINE (BIT 0)
   415  545         1170 C=REGN 9              GET # OF BYTES TO WRITE
   416  546          174 RCR    4              C[X] = # OF BYTES TO WRITE
   417  547          160 N=C                   N[X] = # OF BYTES TO WRITE
   418  550          174 RCR    4              C[3:0] = PROGRAM HEAD ADDR
   419  551           34 PT=    3              POINT AT LOW DIGIT OF C[M]
   420  552          412 A=C    WPT            A[3:0] = PROGRAM HEAD ADDR
   421  553           52 B=0    WPT            B[X] = CHECKSUM (CLEARED)
   422  554 WRTP70   260 C=N                   C = # OF BYTES TO WRITE
   423  555         1146 C=C-1  X              ALL DONE ?
   424  556          157 GOC    WRTP80 ( 573) YES, WRAP UP THE FILE
   425  557          160 N=C                   N = # OF BYTES TO WRITE
   426  560            1 GOSUB  NXBYTA         GET THE NEXT BYTE OF RAM
   426  561            0                       *MAINFRAME: CN10, @0671
   427  562          146 AB EX  X              MOVE CHECKSUM TO REG. A
   428  563          506 A=A+C  X              UPDATE CHECKSUM
   429  564          146 AB EX  X              SAVE CHECKSUM TO REG. B
   430  565            1 GOSUB  SDATA          SEND THE BYTE OF DATA
   430  566            0                       *ILCAS&CTL:  CS0, @0450
   431  567         1114 ?S9=1                 ANY ERROR SO FAR ?
   432  570         1643 GONC   WRTP70 ( 554) NO, WRITE NEXT BYTE
   433  571            1 GOLONG PILERR         YES, ABORT: "TRANSMIT ERR"
   433  572            2                       *ILCAS&CTL:  CS1, @1751
   434  573 WRTP80   306 C=B    X              RETRIEVE THE CHECKSUM
   435  574          144 HPL=CH 1              WRITE CONTROL INT REGISTER
   436  575          405 CH=    @101           END CLASS (BIT 6)
   437  576            1 GOSUB  SDATA          SEND CHECKSUM BYTES
   437  577            0                       *ILCAS&CTL:  CS0, @0450
   438  600            1 GOLONG CSCKUT         CHECK STATUS AND UNTALK
   438  601            2                       *ILCAS&CTL:  CS1, @0654
*
*****************************************************************
* READP  - READ PROGRAM FROM A PROGRAM FILE ON THE CASSETTE     *
* READSB - READ SUBROUTINE. SAME AS READP EXCEPT ALWAYS APPENDS *
*          THE PROGRAM TO END OF MEMORY (USER FUNCTION IS "READSUB")
*                                                               *
* INPUT:   ALPHA REGISTER = FILE NAME                           *
*          READP WILL HONOR THE KEY REASSIGNMENT WITH THE ALPHA LABEL
*          IF THE PROGRAM IS READ IN USER MODE                  *
*****************************************************************
*
   450                   ENTRY  READP
   451                   ENTRY  READSB
*
   453  602          202 CON    @202           B
   454  603           25 CON    @25            U
   455  604           23 CON    @23            S
   456  605            4 CON    @04            D
   457  606            1 CON    @01            A
   458  607            5 CON    @05            E
```

```
  459  610          22 CON    @22           R
  460  611 READSB  604 S11=   0             REMEMBER THIS IS READSUB
  461  612          73 GOTO   RP100  ( 621) LOOK FOR PROGRAM FILE
*
  463  613         220 CON    @220          P
  464  614           4 CON    @04           D
  465  615           1 CON    @01           A
  466  616           5 CON    @05           E
  467  617          22 CON    @22           R
  468  620 READP   610 S11=   1             REMEMBER THIS IS READP
  469  621 RP100   460 LDI                  LOAD LOW 12 BITS OF C WITH
  470  622          10 CON    8             TYPE 8 IS PROGRAM FILE
  471  623           1 GOSUB  FLSCH0        LOOK FOR THE PRGM FILE
  471  624           0                      *ILCAS&CTL:  CS2, @0012
  472  625         614 ?S11=1               IS THIS READSUB ?
  473  626          57 GOC    RP150  ( 633) NO, IT IS READP
  474  627           1 GOSUB  GTFEND        GET FINAL END ADDRESS
  474  630           0                      *MAINFRAME:  CN8, @0350
  475  631           2 A=0    PT            A[3:0] = FINAL END ADDR
  476  632         313 GOTO   RP245  ( 663) PUT PROGRAM AFTER .END.
  477  633 RP150  1314 ?S13=1               RUNNING ?
  478  634          57 GOC    RP200  ( 641) YES, SEE IF IN LAST PRGM
  479  635        1670 C=REGN 14             NO, READ FLAGS REGISTER
  480  636        1530 ST=C                  ST = FLAGS 48-55
  481  637         114 ?S4=1                 SINGLE STEP (FLAG 51) ?
  482  640         173 GONC   RP240  ( 657) NO, CHANGE PC
  483  641 RP200   314 ?S10=1                ARE WE IN ROM ?
  484  642         147 GOC    RP220  ( 656) YES, DON'T CHANGE PC
  485  643           1 GOSUB  FLINKP        NO, FIND PROGRAM END
  485  644           0                      *MAINFRAME: CN10, @0445
  486  645         474 RCR    8             C[3:0] = CURRENT PRGM END
  487  646         412 A=C    WPT           A[3:0] = CURRENT PRGM END
  488  647           1 GOSUB  INCADA        INCREMENT ADDRESS IN RAM
  488  650           0                      *MAINFRAME: CN10, @0726
  489  651           1 GOSUB  NXBYTA        GET 3RD BYTE OF "END"
  489  652           0                      *MAINFRAME: CN10, @0671
  490  653        1730 CST EX               C[1:0] INTO STATUS FLAGS
  491  654         214 ?S5=1                 IS THIS THE FINAL END ?
  492  655          27 GOC    RP240  ( 657) YES, CHANGE PRGM COUNTER
  493  656 RP220   604 S11=   0             REMEMBER DON'T CHANGE PC
  494  657 RP240     1 GOSUB  GTFEND        GET FINAL END ADDRESS
  494  660           0                      *MAINFRAME:  CN8, @0350
  495  661           1 GOSUB  CPGM10        GET CURRENT PROGRAM HEAD
  495  662           0                      *MAINFRAME:  CN1, @1177
  496  663 RP245   212 B=A    WPT           B[3:0] = STARTING ADDRESS
  497  664           1 GOSUB  MEMLFT        COMPUTE AVAILABLE REGS
  497  665           0                      *MAINFRAME:  CN1, @0641
  498  666         406 A=C    X             A[X]=# OF UNUSED MEM REGS
  499  667         116 C=0                  CLEAR ACCUMULATOR
  500  670        1160 DADD=C               ENABLE CHIP 0
  501  671         312 C=B    WPT           C[3:0] = STARTING ADDRESS
  502  672        1150 REGN=C 9             SAVE START ADDR IN REG.9
  503  673        1570 C=REGN 13            GET .END. ADDRESS
  504  674         246 AC EX  X             A[2:0]=.END. C[2:0]=UNUSED
  505  675         706 A=A-C  X             A[2:0] = ENDING ADDRESS
  506  676           2 A=0    PT            CLEAR DIGIT ABOVE A[X]
  507  677         152 AB EX  WPT           A[X]=START, B[X]=END ADDR
  508  700           1 GOSUB  CNTBYT        COMPUTE TOTAL AVAIL BYTES
  508  701           0                      *ILCAS&CTL:  CS3, @1647
  509  702         260 C=N                  C[X]=PRGM SIZE IN BYTES
```

```
   510  703         1406 ? A<C   X              ENOUGH ROOM ?
   511  704          253 GONC    RP250   ( 731) YES, ENOUGH ROOM TO READ
*
   513                   ENTRY   NORMCK
*
   515  705 NORMCK     1 GOSUB   UNT            SEND UNTALK COMMAND
   515  706            0                        *ILCAS&CTL:  CS0, @0254
   516  707            1 GOSUB   LDSST0         SEE IF SST FLAG SET
   516  710            0                        *MAINFRAME:  CN1, @1627
   517  711         1314 ?S13=1                 RUNNING ?
   518  712           47 GOC     NOROOM  ( 716) YES, DISPLAY "NO ROOM"
   519  713          114 ?S4=1                  SINGLE STEP (FLAG 51) ?
   520  714            1 GOLNC   PACKE          NO, PACK & "TRY AGAIN"
   520  715            2                        *MAINFRAME:  CN8, @0002
*
   522                   ENTRY   NOROOM
*
   524  716 NOROOM     1 GOSUB   PLEREX         PIL ERROR EXIT: NO ROOM
   524  717            0                        *ILCAS&CTL:  CS1, @1663
   525  720           16 CON     @16            N
   526  721           17 CON     @17            O
   527  722           40 CON     @40                 BLANK
   528  723           22 CON     @22            R
   529  724           17 CON     @17            O
   530  725           17 CON     @17            O
   531  726         1015 CON     @1015          M
   532  727            1 GOLONG CSEREX          CASSETTE ERROR EXIT
   532  730            2                        *ILCAS&CTL:  CS3, @0376
   533  731 RP250      1 GOSUB   SEEKRN         SEEK TO THE FILE & READ IT
   533  732            0                        *ILCAS&CTL:  CS3, @1567
   534  733            1 GOSUB   SNDATA         SEND SEC. CMD - SEND DATA
   534  734            0                        *ILCAS&CTL:  CS0, @0346
   535  735         1170 C=REGN 9               GET STARTING ADDRESS
   536  736           16 A=0                    CLEAR REGISTER A
   537  737          412 A=C     WPT            A[3:0] = STARTING ADDRESS
   538  740          260 C=N                    C[X] = PRGM SIZE IN BYTES
   539  741          530 M=C                    SAVE PROGRAM SIZE TO REG M
   540  742 RP300    630 C=M                    GET PROGRAM SIZE FROM M
   541  743         1146 C=C-1   X              DECREMENT C, ALL DONE ?
   542  744          217 GOC     RP320   ( 765) YES, GET LAST BYTE
   543  745          530 M=C                    NO, SAVE PROGRAM SIZE TO M
   544  746            1 GOSUB   RDDFRM         READ NEXT FRAME
   544  747            0                        *ILCAS&CTL:  CS0, @0420
   545  750         1114 ?S9=1                  ANY ERROR FOUND ?
   546  751          277 GOC     RP330   (1000) YES, ABORT READING PROGRAM
   547  752         1200 HPIL=C 2               NO, ECHO FRAME JUST READ
   548  753          256 AC EX                  SWAP REGISTERS A AND C
   549  754          174 RCR     4              C[X] = RUNNING CHECKSUM
   550  755         1006 C=A+C   X              ADD FRAME TO CHECKSUM
   551  756          374 RCR     10             ROTATE BACK INTO PLACE
   552  757          256 AC EX                  SWAP A & C BACK AS BEFORE
   553  760            1 GOSUB   INCADA         POINT TO NEXT BYTE
   553  761            0                        *MAINFRAME: CN10, @0726
   554  762            1 GOSUB   PTBYTA         STORE THE BYTE
   554  763            0                        *MAINFRAME:  CN8, @1443
   555  764         1563 GOTO    RP300   ( 742) READ NEXT PROGRAM BYTE
*
   557  765 RP320    206 B=A     X              B[X] = LAST BYTE ADDRESS
   558  766            1 GOSUB   NRD            READ LAST BYTE AS CHECKSUM
   558  767            0                        *ILCAS&CTL:  CS0, @0364
```

```
  559  770         256 AC EX                GET CHECKSUM INFORMATION
  560  771         174 RCR    4             C[X] = CHECKSUM
  561  772         126 C=0    XS            CLEAR C[2] EXPONENT SIGN
  562  773        1546 ? A#C  X             CHECKSUM MATCH ?
  563  774         323 GONC   RP400  (1026) YES, NO ERROR, CONTINUE
  564  775        1110 S9=    1             NO, SET ERROR FLAG
  565  776         404 S8=    0             S8 CLEAR = CHECKSUM ERROR
  566  777          33 GOTO   RP335  (1002) CLEAN UP AFTER ERROR
  567 1000 RP330   410 S8=    1             S8 SET = NON-CHKSUM ERROR
  568 1001         206 B=A    X             B[X] = LAST BYTE ADDRESS
* ERROR OCCURRED, THE PROGRAM HAS NOT BEEN ALL READ INTO MEMORY.
* GENERATE AN "END" AT THE BEGINNING OF LOADING SPACE AND CLEAN
* MEMORY FROM THE "END" TO LAST BYTE LOADING ADDRESS.
  572                ENTRY  RP335
  573 1002 RP335   116 C=0                 CLEAR ACCUMULATOR
  574 1003        1160 DADD=C              ENABLE CHIP 0
  575 1004        1170 C=REGN 9            GET STARTING ADDRESS
  576 1005         412 A=C    WPT           SAVE STARTING ADDRESS TO A
  577 1006         316 C=B                  GET LAST BYTE ADDRESS IN C
  578 1007         530 M=C                  SAVE LAST BYTE ADDR INTO M
  579 1010           1 GOSUB  CLTAIL        CLEAR TRAILING BYTE IN REG
  579 1011           0                      *ILCAS&CTL:  CS2, @1310
  580 1012         246 AC EX  X             STARTING ADDRESS TO REG C
  581 1013        1146 C=C-1  X             DECREMENT STARTING ADDRESS
  582 1014        1160 DADD=C              POINT TO BEG OF LOAD SPACE
  583 1015         406 A=C    X             SAVE BEG LOAD SPACE TO A
  584 1016         116 C=0                  CLEAR ACCUMULATOR
  585 1017         234 PT=    5             POINT AT DIGIT 5
  586 1020        1420 LC     12            C = "END" (00000000C00000)
  587 1021        1360 DATA=C              WRITE END AT BEG LOAD SPACE
  588 1022         630 C=M                  GET LAST LOADING ADDRESS
  589 1023         246 AC EX  X             A[X]=LLA, C[X]=ADDR NEW END
  590 1024           1 GOSUB  CLM10         CLEAR THE MEMORY
  590 1025           0                      *ILCAS&CTL:  CS2, @1277
  591                ENTRY  RP400
  592 1026 RP400     1 GOSUB  UNT           SEND UNTALK COMMAND
  592 1027           0                      *ILCAS&CTL,  CS0, @0254
  593                ENTRY  RP401
  594 1030 RP401   116 C=0                 CLEAR ACCUMULATOR
  595 1031         530 M=C                  CLEAR REGISTER M
  596 1032        1160 DADD=C              ENABLE CHIP 0
  597 1033        1170 C=REGN 9            GET STARTING ADDRESS
  598 1034         416 A=C                  SAVE START ADDR TO REG A
  599 1035          34 PT=    3             POINT AT LOWEST 2 BYTES
  600 1036         614 ?S11=1               RUNNING OR SINGLE STEP ?
  601 1037          43 GONC   RP403  (1043) YES, DON'T CHANGE PC
  602 1040         304 S10=   0             NO, CLEAR ROM FLAG
  603 1041           1 GOSUB  PUTPCX        SET PC TO BEGINNING OF PGM
  603 1042           0                      *MAINFRAME:  CN8, @1457
  604 1043 RP403  1570 C=REGN 13           GET ADDRESS OF REGISTER 0
  605 1044          74 RCR    3             C[2:0] = ADDRESS OF REG 0
  606 1045        1546 ? A#C  X             STARTING FROM REG.0 ?
  607 1046         163 GONC   RP410  (1064) YES, NO NEED TO CHG END
  608 1047           1 GOSUB  GTBYTA        NO, CHG PREV. FINAL END
  608 1050           0                      *MAINFRAME: CN10, @0673
  609 1051        1730 CST EX                 TO LOCAL END
  610 1052         204 S5=    0             CHANGE FINAL TO LOCAL END
  611 1053        1730 CST EX              BRING CHANGED BYTE INTO C
  612 1054           1 GOSUB  PTBYTA        WRITE BACK AS LOCAL END
  612 1055           0                      *MAINFRAME:  CN8, @1443
```

```
   613 1056           1 GOSUB  DECADA         DECREMENT ADDRESS IN RAM
   613 1057           0                       *MAINFRAME: CN10, @0712
   614 1060           1 GOSUB  DECADA         POINT TO 1ST OF PREV. END
   614 1061           0                       *MAINFRAME: CN10, @0712
   615 1062         256 AC EX                 GET PREV LINK ADDR INTO C
   616 1063         530 M=C                   SAVE PREV LINK ADDR IN M
   617 1064 RP410   116 C=0                   CLEAR ACCUMULATOR
   618 1065        1160 DADD=C                ENABLE CHIP 0
   619 1066        1170 C=REGN 9              GET STARTING ADDRESS
   620 1067         416 A=C                   SAVE START ADDRESS INTO A
   621 1070        1670 C=REGN 14             READ FLAG REGISTER
   622 1071        1274 RCR    7              C[1:0] = FLAGS 20-27
   623 1072        1530 ST=C                  SAVE FLAGS TO STATUS
   624 1073        1204 S7=    0              CLEAR USER MODE (FLAG 20)
   625 1074         340 SEL Q                 SELECT POINTER Q
   626 1075        1334 PT=    13             Q POINTS TO MANTISSA SIGN
   627 1076         240 SEL P                 SELECT POINTER P
   628 1077        1334 PT=    13             P POINTS TO MANTISSA SIGN
   629 1100        1420 LC     12             C[S] = 12 (C HEX)
   630 1101        1520 LC     13             C[12] = 13 (D HEX)
   631 1102         422 A=C    PQ             A[13:11] = CD0
   632 1103 RP425    34 PT=    3              POINT AT LOWEST 2 BYTES
* START SET LINK HERE - LOOK AT NEXT BYTE TO SEE IF IT IS AN ALBL
*
   635 1104 RP430   212 B=A    WPT            B[3:0] = STARTING ADDRESS
   636 1105           1 GOSUB  NXBYTA         GET THE NEXT BYTE OF RAM
   636 1106           0                       *MAINFRAME: CN10, @0671
   637 1107        1074 RCR    2              C[13:12] = PROGRAM BYTE
   638 1110        1534 PT=    12             SET POINTER P AT C[12]
   639 1111        1362 ? C#0  PQ             C[13:0] IS A NULL ?
   640 1112        1713 GONC   RP425  (1103)  YES, A NULL INSTRUCTION
   641 1113        1576 ? A#C  S              AN END OR ALBL (C0 HEX) ?
   642 1114         417 GOC    RP450  (1155)  NO, INST. NOT FROM ROW C
   643 1115        1402 ? A<C  PT             CE (X<>NN) OR CF (LBL.NN)?
   644 1116         377 GOC    RP450  (1155)  YES, X<>NN OR LBL.NN INST.
   645 1117          34 PT=    3              P POINTS AT LOW 2 BYTES
* SEE IF IT IS AN END
*
   648 1120         212 B=A    WPT            B[3:0] = STARTING ADDRESS
   649 1121           1 GOSUB  INCAD2         INCREMENT ADDRESS 2 BYTES
   649 1122           0                       *MAINFRAME: CN10, @0723
   650 1123           1 GOSUB  GTBYTA         GET NEXT BYTE
   650 1124           0                       *MAINFRAME: CN10, @0673
   651 1125        1574 RCR    12             C[3:2] = BYTE JUST READ
   652 1126         152 AB EX  WPT            A[3:0] = STARTING ADDRESS
   653 1127        1042 C=C+1  PT             IS IT AN END ?
   654 1130         323 GONC   RP470  (1162)  YES, HIGH DIGIT C[3] < F
* RECOMPUTE THE LINK FOR AN ALBL
*
   657 1131         630 C=M                   C = PREVIOUS LINK ADDRESS
   658 1132         252 AC EX  WPT            C[3:0] = STARTING ADDRESS
   659 1133           1 GOSUB  GENLNK         GENERATE A LINK
   659 1134           0                       *MAINFRAME:  CN8, @1632
   660 1135         412 A=C    WPT            A[3:0] = RETURNED LINK
   661 1136         530 M=C                   M = RETURNED LINK ADDRESS
   662 1137        1614 ?S0=1                 ARE WE IN USER MODE ?
   663 1140         127 GOC    RP440  (1152)  YES, DON'T CLEAR KEY CODE
   664 1141           1 GOSUB  INCAD2         INCREMENT ADDRESS 2 BYTES
   664 1142           0                       *MAINFRAME: CN10, @0723
   665 1143           1 GOSUB  INCADA         POINT TO KEY CODE
```

```
   665 1144           0                        *MAINFRAME: CN10, @0726
   666 1145         106 C=0    X               CLEAR C[2:0]
   667 1146           1 GOSUB  PTBYTA          CLEAR THE KEY CODE
   667 1147           0                        *MAINFRAME:  CN8, @1443
   668 1150         630 C=M                    GET LINK INFORMATION TO C
   669 1151         412 A=C    WPT             A[3:0] = LINK ADDRESS
   670 1152 RP440     1 GOSUB  DECADA          POINT TO 1 BYTE BEFORE ALBL
   670 1153           0                        *MAINFRAME: CN10, @0712
   671 1154          33 GOTO   RP460  (1157)   SKIP TO NEXT INSTRUCTION
   672 1155 RP450    34 PT=    3               INST. FOUND NOT C0 TO CD
   673 1156         152 AB EX  WPT             A[3:0] = STARTING ADDRESS
   674 1157 RP460     1 GOSUB  NXLDEL          SKPLIN ENTRY NOT CHK ROMFLG
   674 1160           0                        *MAINFRAME: CN10, @1375
   675 1161        1233 GOTO   RP430  (1104)   GET NEXT BYTE OF RAM
* SET LINK FOR FINAL END AND PUT IT TO PROPER PLACE.
* (FINAL END HAS TO BE RIGHT-JUSTIFIED IN A REGISTER)
*
   679 1162 RP470   202 B=A    PT              SAVE BYTE POINTER IN B[3]
   680 1163           1 GOSUB  DECADA          POINT BACK ONE BYTE
   680 1164           0                        *MAINFRAME: CN10, @0712
   681 1165           1 GOSUB  CLTAIL          CLEAR TRAILING BYTE IN REG
   681 1166           0                        *ILCAS&CTL:  CS2, @1310
   682 1167         142 AB EX  PT              RESTORE BYTE POINTER
   683 1170         420 LC     4               NEED 4 BYTES FOR .END.
   684 1171          34 PT=    3               RESTORE POINTER TO DIGIT 3
   685 1172        1402 ? A<C  PT              CAN .END. BE PUT IN REG ?
   686 1173          37 GOC    RP480  (1176)   NO, PUT .END. TO NEXT REG
   687 1174         402 A=C    PT              SAVE BYTE POINTER TO A[3]
   688 1175          73 GOTO   RP490  (1204)   GENERATE LINK FOR .END.
   689 1176 RP480   246 AC EX  X               C[X] = STARTING ADDRESS
   690 1177        1146 C=C-1  X               PUT .END. TO NEXT REG
   691 1200        1160 DADD=C                 POSITION TO NEXT REGISTER
   692 1201         412 A=C    WPT             A[3]=4, A[2:0]=START ADDR
   693 1202         116 C=0                    CLEAR ACCUMULATOR
   694 1203        1360 DATA=C                 WRITE 0 TO NEXT REGISTER
* GENERATE LINK FOR FINAL END
*
   697 1204 RP490   630 C=M                    GET LINK INFORMATION TO C
   698 1205         252 AC EX  WPT             C[3]=4, C[2:0]=START ADDR
   699 1206           1 GOSUB  GENLNK          GENERATE A LINK
   699 1207           0                        *MAINFRAME:  CN8, @1632
* UPDATE CHAIN HEAD
*
   702 1210         530 M=C                    SAVE NEW .END. ADDR IN M
   703 1211        1160 DADD=C                 POINT TO NEW .END. ADDR
   704 1212         260 C=N                    GET FILE INFORMATION TO C
   705 1213         674 RCR    11              C[2:1] = FILE TYPE
   706 1214        1530 ST=C                   MOVE FILE PROT. TO STATUS
   707 1215          70 C=DATA                 GIVE LAST BYTE TO FINAL END
   708 1216        1434 PT=    1               POINT AT LOWEST BYTE OF C
   709 1217         220 LC     2               C[1] = 2
   710 1220        1520 LC     13              C[1:0] = 2C
* CHECK IF THIS IS A PRIVATE PROGRAM
*
   713 1221         114 ?S4=1                  PRIVATE PROGRAM ?
   714 1222          33 GONC   RP500  (1225)   NO, DON'T SET PRIVATE BIT
   715 1223        1434 PT=    1               POINT AT LOWEST BYTE OF C
   716 1224         620 LC     6               SET PRIVATE BIT C[1:0]=6C
* CLEAR MEMORY BETWEEN OLD FINAL END AND NEW FINAL END
*
```

```
   719 1225 RP500  1360 DATA=C               WRITE 2C OR 6C TO REGISTER
   720 1226         630 C=M                  GET NEW .END. ADDRESS
   721 1227           1 GOSUB  CLNMEM        CLEAN TRAIL MEMORY
   721 1230           0                      *ILCAS&CTL:  CS2, @1271
   722 1231        1004 S2=    0             CALLED FROM READP/READSUB
   723 1232           1 GOSUB  RSTKCA        REBUILD KEY REASSIGNMENT
   723 1233           0                      *ILCAS&CTL:  CS2, @1320
   724 1234        1114 ?S9=1                ANY ERROR FOUND ?
   725 1235         253 GONC   RP550  (1262) NO, CHECK RUNNING OR SST
   726 1236        1304 S13=   0             YES, WE FOUND AN ERROR
   727 1237         414 ?S8=1                WAS IT A CHECKSUM ERROR ?
   728 1240           1 GOLC   CSERCK        NO, SEE WHAT ERROR IT IS
   728 1241           3                      *ILCAS&CTL:  CS1, @1064
   729                ENTRY  CKSUME
   730 1242 CKSUME    1 GOSUB  CKSMER        DISPLAY "READ ERR"
   730 1243           0                      *ILCAS&CTL:  CS2, @1246
   731 1244           1 GOLONG CSERE0        CASSETTE ERROR EXIT
   731 1245           2                      *ILCAS&CTL:  CS3, @0400
   732                ENTRY  CKSMER
*
   734 1246 CKSMER    1 GOSUB  MESSLP        CLEAR LCD & PRINT MESSAGE
   734 1247           0                      *ILCAS&CTL:  CS1, @1667
   735 1250          22 CON    @22           R
   736 1251           5 CON    @05           E
   737 1252           1 CON    @01           A
   738 1253           4 CON    @04           D
   739 1254          40 CON    @40                BLANK
   740 1255           5 CON    @05           E
   741 1256          22 CON    @22           R
   742 1257        1022 CON    @1022         R
   743 1260           1 GOLONG LJDLY         LEFT-JUSTIFY & DELAY
   743 1261           2                      *ILCAS&CTL:  CS1, @1463
   744 1262 RP550   614 ?S11=1               RUNNING OR SST ?
   745 1263          43 GONC   RP560  (1267) YES, NORMAL RTN W/PUSH
   746 1264         214 ?S5=1                AUTO RUN PROGRAM ?
   747 1265           1 GOLC   WKUP80        YES, RUN AND SOUND A BEEP
   747 1266           3                      *MAINFRAME:  CN0, @0777
   748 1267 RP560     1 GOLONG NFRPU         NORMAL FUNC RETURN W/PUSH
   748 1270           2                      *MAINFRAME:  CN0, @0360
*****************************************************************
* CLNMEM - AFTER READING IN A PROGRAM, CLEAN THE SPACE BETWEEN  *
*          THE OLD FINAL END AND THE NEW FINAL END              *
*          CALL WITH NEW FINAL END ADDRESS IN C[2:0]            *
*                                                               *
* CLM10  - SPECIAL ENTRY                                        *
*          CLEAR MEMORY FROM ADDR IN C[X] (LARGER)              *
*                        TO ADDR IN A[X] (SMALLER)              *
*****************************************************************
   758                ENTRY  CLNMEM
   759                ENTRY  CLM10
*
   761 1271 CLNMEM  406 A=C    X            A[X] = NEW FINAL END ADDR
   762 1272         106 C=0    X            CLEAR C[X]
   763 1273        1160 DADD=C              ENABLE CHIP 0
   764 1274        1570 C=REGN 13           GET OLD FINAL END ADDRESS
   765 1275         246 AC EX  X            C[X] = NEW FINAL END ADDR
   766 1276        1550 REGN=C 13           UPDATE NEW FINAL END ADDR
   767 1277 CLM10    56 B=0                 CLEAR B-REGISTER
   768 1300 CLM15  1406 ? A<C  X            ARE WE DONE YET ?
   769 1301        1640 RTN NC              YES, SUCCESSFUL RETURN
```

```
   770 1302        1146 C=C-1  X             NO, BACK UP ONE REGISTER
   771 1303        1160 DADD=C               POINT AT THAT REGISTER
   772 1304         356 BC EX                SAVE POINTER, C NOW ZERO
   773 1305        1360 DATA=C               WRITE ZEROS TO REGISTER
   774 1306         356 BC EX                C HAS POINTER, B IS ZERO
   775 1307        1713 GOTO   CLM15  (1300) CLEAR ANOTHER REGISTER
*************************************************************************
* CLTAIL - CLEAR TRAILING BYTE IN A REGISTER                            *
* INPUT:   A[3:0] = ADDRESS ONE BYTE PRECEDING STARTING BYTE            *
* USES:    C, B[1:0], +1 SUB LEVEL                                      *
*************************************************************************
   781                  ENTRY  CLTAIL
*
   783 1310 CLTAIL    1 GOSUB  INCADA        INCREMENT ADDRESS IN RAM
   783 1311           0                      *MAINFRAME: CN10, @0726
   784 1312         106 C=0    X             CLEAR LOWEST 3 DIGITS OF C
   785 1313           1 GOSUB  PTBYTA        STORE 0 AT ADDRESS A[3:0]
   785 1314           0                      *MAINFRAME:  CN8, @1443
   786 1315        1502 ? A#0  PT            ARE WE DONE YET ?
   787 1316        1727 GOC    CLTAIL (1310) NO, CLEAR ANOTHER BYTE
   788 1317        1740 RTN                  YES, RETURN TO CALLER
   789                  FILLTO @1317
*************************************************************************
* RSTKCA - REBUILD THE KEY REASSIGNMENT BIT MAPS                        *
*          AFTER READING IN THE STATUS TRACKS, OLD KEY REASSIGNMENTS    *
*          ARE DESTROYED AND MUST BE REBUILT. IN THIS CASE, THE KEY     *
*          REASSIGNMENT JUST READ WILL TAKE PRECEDENCE.  READING IN     *
*          A PROGRAM WILL DESTROY THE KEY CODE USED IN LAST PROGRAM.    *
*          IF READING IN PROGRAM IN USER MODE, THE KEY ASSIGNED TO      *
*          ANY ALBL IN THE PROGRAM JUST READ WANTS TO TAKE PRECEDENCE.  *
*          THEREFORE, AFTER READING A PROGRAM, THE KEY REASSIGNMENT     *
*          BIT MAPS MUST BE REBUILT. THE PROCEDURES ARE AS FOLLOWS:     *
*          1. CLEAR THE ENTIRE BIT MAP                                  *
*          2. RESTORE KEY REASSIGNMENT OF HARDCODE & XROM FUNCTIONS     *
*          3. RESTORE KEY REASSIGNMENT IN ALBL, IF THE KEY HAS BEEN     *
*             ALREADY ASSIGNED TO ANOTHER FUNCTION :                    *
*            A. IF AFTER READING STATUS TRACK, CLEAR KEY CODE IN ALBL.  *
*            B. IF AFTER READING PROGRAM, FIND THE KEY CODE SOMEWHERE    *
*               ELSE AND CLEAR IT THERE.                                *
*                                                                       *
*          CALL WITH S2 = 1 MEANS FROM R/STS (READ STATUS TRACK)        *
*                   = 0 MEANS FROM R/PGM (READ PROGRAM FILE)            *
*************************************************************************
   811                  ENTRY  RSTKCA
*
   813 1320 RSTKCA    1 GOSUB  ENCP00        ENABLE CHIP 0
   813 1321           0                      *MAINFRAME:  CN2, @0522
   814 1322        1770 C=REGN 15            GET SHIFTED KEY ASSIGNS
   815 1323         132 C=0    M             CLEAR SHIFTED KEY ASSIGNS
   816 1324         136 C=0    S             CLEAR REMAINING SK ASSIGNS
   817 1325        1750 REGN=C 15            WRITE CLEARED SK REGISTER
   818 1326         460 LDI                  LOAD LOW 12 BITS OF C WITH
   819 1327         277 CON2   11     15     BF = REG 00 (ADDR: C0) - 1
*************************************************************************
* GET THE KEY CODE FROM KEY REASSIGNMENT RECORD AND SET ITS BIT         *
* IN THE BIT MAP.                                                       *
*************************************************************************
   824 1330 RTKC10 1046 C=C+1  X             POINT TO REG, 1ST ADDR: C0
   825 1331        1204 S7=    0             NOT DONE WITH 1ST KEY CODE
   826 1332        1250 REGN=C 10            WRITE UNSHIFTED KA BIT MAP
```

```
827 1333          416 A=C                    UNSHIFTED KA BIT MAP TO A
828 1334         1570 C=REGN 13              GET ADDRESS OF .END.
829 1335          256 AC EX                  A[X]=.END., C[X]=CUR KA REG
830 1336         1546 ? A#C  X               REACHED CHAIN HEAD ?
831 1337          313 GONC   RTKC30 (1370)   YES, DONE WITH ALL KA REGS
832 1340 RTKC15  1160 DADD=C                SET DATA PTR TO CUR KA REG
833 1341           70 C=DATA                READ KEY ASSIGNMENT REG
834 1342          416 A=C                   SAVE KA REG TO REGISTER A
835 1343          106 C=0    X              CLEAR C[2:0]
836 1344         1160 DADD=C                ENABLE CHIP 0
837 1345          256 AC EX                 RESTORE KA REGISTER TO C
838 1346         1076 C=C+1  S              STILL A KEY REASSIGN REG ?
839 1347          213 GONC   RTKC30 (1370)  NO, DONE WITH ALL THE REGS
840 1350         1434 PT=    1              POINT AT LOWEST BYTE
841 1351         1214 ?S7=1                 DONE WITH FIRST KEY CODE ?
842 1352           23 GONC   *+2    (1354)  YES, DON'T ROTATE REGISTER
843 1353          574 RCR    6              NO, ROTATE TO READ 1ST KC
844 1354         1352 ? C#0  WPT            IS THERE A KEY CODE ?
845 1355           63 GONC   RTKC20 (1363)  NO, READ UNSHIFTED KA MAP
846 1356          416 A=C                   YES, SAVE KA REGISTER TO A
847 1357            1 GOSUB  TBITMA         FIND KC BIT IN BIT MAP
847 1360            0                       *MAINFRAME: CN11, @1577
848 1361            1 GOSUB  SRBMAP         SET KC BIT IN BIT MAP
848 1362            0                       *MAINFRAME: CN11, @1645
849 1363 RTKC20  1270 C=REGN 10             READ UNSHIFTED KEY ASSIGNS
850 1364         1214 ?S7=1                 DONE WITH 2ND KEY CODE ?
851 1365         1437 GOC    RTKC10 (1330)  YES, READ NEXT KA REGISTER
852 1366         1210 S7=    1              NO, SET FLAG AFTER 1ST KC
853 1367         1513 GOTO   RTKC15 (1340)  READ SECOND CODE IN KA REG
854 1370 RTKC30  1014 ?S2=1                 CALLED FROM R/STS ?
855 1371          137 GOC    RTKC40 (1404)  YES, SKIP NEXT 10 BYTES
856 1372         1170 C=REGN 9              GET START LOADING ADDRESS
857 1373           34 PT=    3              POINT AT LOWEST 2 BYTES
858 1374          412 A=C    WPT            A[3:0] = START LOADING ADR
859 1375            1 GOSUB  DECADA         DECREMENT ADDRESS IN RAM
859 1376            0                       *MAINFRAME: CN10, @0712
860 1377            1 GOSUB  DECADA         POINT TO PREVIOUS END ADDR
860 1400            0                       *MAINFRAME: CN10, @0712
861 1401         1270 C=REGN 10             READ UNSHIFTED KA BIT MAP
862 1402          252 AC EX  WPT            C[3:0] = PREV END ADDRESS
863 1403         1250 REGN=C 10             SAVE PE ADDRESS IN REG.10
864 1404 RTKC40     1 GOSUB  GTFEND        GET FINAL END ADDRESS
864 1405            0                       *MAINFRAME:  CN8, @0350
865 1406 RTKC45    34 PT=    3              POINT TO LOWEST 2 BYTES
866 1407            1 GOSUB  GTLINK         GET A LINK FROM ADDR IN A
866 1410            0                       *MAINFRAME:  CN8, @1116
867 1411         1346 ? C#0  X              CHAIN END ?
868 1412            1 GOLNC  ENCP00         YES, ALL DONE. RETURN
868 1413            2                       *MAINFRAME:  CN2, @0522
869 1414            1 GOSUB  UPLINK         NO, MOVE UP 1 LABEL LINK
869 1415            0                       *MAINFRAME:  CN8, @1065
870 1416         1076 C=C+1  S              IS IT AN END ?
871 1417         1673 GONC   RTKC45 (1406)  YES, KEEP SEEKING .END.
872 1420          106 C=0    X              NO, IT'S AN ALPHA LABEL
873 1421         1160 DADD=C                ENABLE CHIP 0
874 1422          252 AC EX  WPT            C[3:0]=LINK AND ADDRESS
875 1423         1150 REGN=C 9              SAVE LINK & ADDR IN REG.9
876 1424          412 A=C    WPT            A[3:0]=LINK AND ADDRESS
877 1425            1 GOSUB  INCAD2         POINT TO KEY CODE OF ALBL
877 1426            0                       *MAINFRAME: CN10, @0723
```

```
   878 1427           1 GOSUB  NXBYTA        GET KEY CODE OF ALBL
   878 1430           0                      *MAINFRAME: CN10, @0671
   879 1431         252 AC EX  WPT           C[3:0]=LINK AND ADDRESS
   880 1432         160 N=C                  SAVE LINK AND ADDRESS IN N
   881 1433         106 C=0    X             CLEAR C[2:0]
   882 1434        1160 DADD=C               ENABLE CHIP 0
   883 1435        1570 C=REGN 13            READ FINAL END ADDRESS
   884 1436         530 M=C                  SAVE FINAL END ADDR INTO M
   885 1437          26 A=0    XS            CLEAR A[2] EXPONENT SIGN
   886 1440        1506 ? A#0  X             IS THERE A KEY CODE ?
   887 1441         323 GONC   RUSR40 (1473) NO KEY CODE WITH THIS ALBL
   888 1442         206 B=A    X             SAVE KEY CODE TO B[X]
   889 1443           1 GOSUB  TBITMA        TEST THE BIT MAP
   889 1444           0                      *MAINFRAME: CN11, @1577
   890 1445        1356 ? C#0                IS THIS BIT SET ?
   891 1446         233 GONC   RUSR30 (1471) NO, JUST SET IT
   892 1447        1014 ?S2=1                CALLED FROM R/STS ?
   893 1450         113 GONC   RUSR25 (1461) NO, CALLED FROM R/PGM
   894 1451         260 C=N                  GET ADDRESS OF KEY CODE
   895 1452         416 A=C                  SAVE IT INTO REGISTER A
   896 1453         106 C=0    X             CLEAR C[2:0]
   897 1454           1 GOSUB  PTBYTA        CLEAR THE KEY CODE
   897 1455           0                      *MAINFRAME:  CN8, @1443
   898 1456         106 C=0    X             CLEAR C[2:0]
   899 1457        1160 DADD=C               ENABLE CHIP 0
   900 1460         133 GOTO   RUSR40 (1473) GET SAVED LINK & ADDRESS
   901 1461 RUSR25  146 AB EX  X             A[1:0]=KC, CLEAR IT NOW
   902 1462        1270 C=REGN 10            GET UNSHIFTED KC BIT MAP
   903 1463         374 RCR    10            C[7:4]=PREV END ADDRESS
   904 1464         356 BC EX                B[7:4]=PREV END ADDRESS
   905 1465        1410 S1=    1             S1=1 TO CLEAR KEY CODE
   906 1466           1 GOSUB  GCPKC0        CLEAR KEY CODE FROM MAP
   906 1467           0                      *MAINFRAME: CN10, @1611
   907 1470          33 GOTO   RUSR40 (1473) GET SAVED LINK & ADDRESS
   908 1471 RUSR30    1 GOSUB  SRBMAP        SET A BIT IN THE BIT MAP
   908 1472           0                      *MAINFRAME: CN11, @1645
   909 1473 RUSR40 1170 C=REGN 9             GET START LOADING ADDRESS
   910 1474         416 A=C                  SAVE IT TO REGISTER A
   911 1475        1113 GOTO   RTKC45 (1406) SEARCH THE LABEL CHAIN
*
*****************************************************************
* WRTS  -  WRITE STATUS                                         *
*          STATUS INCLUDES :                                    *
*          1. X, Y, Z, T, LASTX AND ALPHA REGISTERS             *
*          2. FLAGS 0-43                                        *
*          3. SIZE AND SIGMA REGISTER ADDRESS                   *
*****************************************************************
*
   921                 ENTRY  WRTS
*
   923 1476         223 CON    @223          S
   924 1477          24 CON    @24           T
   925 1500          22 CON    @22           R
   926 1501          27 CON    @27           W
   927 1502 WRTS    136 C=0    S             CODE 0 = SEARCH FOR FILE
   928 1503           1 GOSUB  FLSCH         SEARCH FOR DUPLICATE FILE
   928 1504           0                      *ILCAS&CTL:  CS2, @0013
   929 1505         460 LDI                  LOAD LOW 12 BITS OF C WITH
   930 1506           6 CON    6             TYPE 6 IS STATUS FILE
   931 1507           1 GOSUB  RWCHK         PURGE IF SAME FILE FOUND
```

```
   931 1510           0                       *ILCAS&CTL:  CS2, @0225
   932 1511         116 C=0                   CLEAR ACCUMULATOR
   933 1512         460 LDI                   START FROM REGISTER 0 AND
   934 1513          12 CON     10            WRITE NEXT 10 REGISTERS
   935 1514           1 GOSUB   RG-BY#        COMPUTE # OF BYTES NEEDED
   935 1515           0                       *ILCAS&CTL:  CS2, @1761
   936 1516        1434 PT=     1             POINT AT LOWEST BYTE
   937 1517         620 LC      6             TYPE 6 IS STATUS FILE
   938 1520           1 GOSUB   CRTFL0        CREATE A FILE ENTRY IN DIR
   938 1521           0                       *ILCAS&CTL:  CS1, @1256
   939 1522           1 GOSUB   SEKSUB        SEEK TO REC & SET WRT MODE
   939 1523           0                       *ILCAS&CTL:  CS0, @0326
   940 1524        1670 C=REGN 14             READ FLAGS REGISTER
   941 1525        1150 REGN=C 9              MOVE REG.14 TO REG.9
*************************************************************************
* COMPUTE SIZE AND STORE IT TO REG.8[13:11]                             *
* COMPUTE RELATIVE POSITION OF SIGMA REG TO REG.0 AND STORE IT          *
* TO REG.8[10:8]                                                        *
*************************************************************************
   947 1526           1 GOSUB   FNDEND        FIND TOP END OF MEMORY
   947 1527           0                       *MAINFRAME:  CN5, @1460
   948 1530         116 C=0                   CLEAR ACCUMULATOR
   949 1531        1160 DADD=C                ENABLE CHIP 0
   950 1532        1570 C=REGN 13             READ SIGMA, R00, .END. REG
   951 1533          74 RCR     3             C[X]=REG0, C[10:8]=SIG REG
   952 1534         706 A=A-C   X             A[X]=# OF DATA REGS(=SIZE)
   953 1535         256 AC EX                 A[X]=REG.0
   954 1536         674 RCR     11            C[5:3]=SIZE,  A[X]=REG.0
   955 1537         272 AC EX   M             A[5:3]=SIZE,  A[X]=REG.0
   956 1540         474 RCR     8             C[X]=SIG REG, A[X]=REG.0
   957 1541         246 AC EX   X             A[X]=SIG REG, C[X]=REG.0
   958 1542         706 A=A-C   X             A[X]=REL POS OF SIGMA REG
   959 1543         234 PT=     5             PUT SIZE & REL.POS. OF SIG
   960 1544        1070 C=REGN 8               TO R.8[13:11] & R.8[10:8]
   961 1545         474 RCR     8             C[5:0] = REG.8[13:8]
   962 1546         252 AC EX   WPT           C[5:0] = SIZE & REL. SIGMA
   963 1547         574 RCR     6             ROTATE BACK TO NORMAL POSN
   964 1550        1050 REGN=C 8              REWRITE REG.8 W/NEW INFO
   965 1551         116 C=0                   CLEAR ACCUMULATOR
   966 1552         460 LDI                   READY TO READ 10 REGISTERS
   967 1553          12 CON     10            C[1:0] = 10 (0A HEX)
   968 1554         530 M=C                   SAVE COUNTER TO REGISTER M
   969 1555          56 B=0                   INITIALIZE CHECKSUM
   970 1556           1 GOLONG WRTA20         WRITE 10 REGS & CHECKSUM
   970 1557           2                       *ILCAS&CTL:  CS3, @0571
*
*************************************************************************
* READS - READ STATUS FILE                                              *
* NOTE:   WILL RESIZE THE MACHINE AND WIPE OUT USER RETURN STACK!!      *
*************************************************************************
*
   977                ENTRY   READS
*
   979 1560         223 CON     @223          S
   980 1561           4 CON     @04           D
   981 1562           1 CON     @01           A
   982 1563           5 CON     @05           E
   983 1564          22 CON     @22           R
   984 1565 READS   460 LDI                   LOAD LOW 12 BITS OF C WITH
   985 1566           6 CON     6             TYPE 6 IS STATUS FILE
```

```
   986 1567           1 GOSUB  FLSCH0         SEARCH THE STATUS FILE
   986 1570           0                       *ILCAS&CTL:  CS2, @0012
   987 1571           1 GOSUB  SEEKRN         READ THAT RECORD
   987 1572           0                       *ILCAS&CTL:  CS3, @1567
   988 1573         116 C=0                   CLEAR ACCUMULATOR
   989 1574         460 LDI                   READ 10 REGISTERS AND
   990 1575          12 CON    10              SAVE THEM IN REG.0-REG.9
   991 1576         530 M=C                   SAVE COUNTER IN REGISTER M
   992 1577          56 B=0                   INITIALIZE CHECKSUM
   993 1600           1 GOSUB  RDREG           READ REGISTERS ROUTINE
   993 1601           0                       *ILCAS&CTL:  CS1, @1020
   994 1602           1 GOSUB  RDRGA           READ CHECKSUM
   994 1603           0                       *ILCAS&CTL:  CS1, @1076
   995 1604         160 N=C                   SAVE CHECKSUM IN N
   996 1605         674 RCR    11             ROTATE BACK AFTER RDRGA
   997 1606           1 GOSUB  NRDC            SEND NOT READY FOR DATA
   997 1607           0                       *ILCAS&CTL:  CS0, @0366
   998 1610           1 GOSUB  UNT             SEND UNTALK COMMAND
   998 1611           0                       *ILCAS&CTL,  CS0, @0254
   999 1612         260 C=N                   GET CHECKSUM FROM N
  1000 1613         156 AB EX                 COPY CHECKSUM B TO A
  1001 1614        1556 ? A#C                 CHECKSUM MATCH ?
  1002 1615           1 GOLC   CKSUME         NO, DISPLAY "READ ERR"
  1002 1616           3                       *ILCAS&CTL,  CS2, @1242
* UPDATE FLAGS 0-43, SIZE AND SIGMA REGISTER
*
  1005 1617        1170 C=REGN 9              LOAD THE FLAGS FROM REG.9
  1006 1620         416 A=C                   COPY FLAGS TO REGISTER A
  1007 1621        1670 C=REGN 14             READ FLAGS REGISTER
  1008 1622         272 AC EX  M              C[12:3] = FLAGS 4-43
  1009 1623         276 AC EX  S              C[13] = FLAGS 0-3
  1010 1624        1650 REGN=C 14             WRITE FLAGS REGISTER
  1011 1625        1070 C=REGN 8              REG.8[13:11] = SIZE
  1012 1626         674 RCR    11             C[X] = SIZE
  1013 1627        1104 S9=    0              CLEAR ERROR FLAG
  1014 1630           1 GOSUB  SIZSUB         TRY TO UPDATE SIZE
  1014 1631           0                       *MAINFRAME:  CN5, @1627
  1015 1632        1114 ?S9=1                 DID WE MAKE IT ?
  1016 1633         113 GONC   RSTS35 (1644)  YES, UPDATE SIGMA REG ADDR
  1017 1634           1 GOSUB  PLEREX         NO, PIL ERROR EXIT: SIZE
  1017 1635           0                       *ILCAS&CTL:  CS1, @1663
  1018 1636          23 CON    @23            S
  1019 1637          11 CON    @11            I
  1020 1640          32 CON    @32            Z
  1021 1641        1005 CON    @1005          E
  1022 1642           1 GOLONG DSPERR         DISPLAY MESSAGE: ERR
  1022 1643           2                       *ILCAS&CTL:  CS3, @0405
  1023 1644 RSTS35 1070 C=REGN 8              REG.8[10:8]=REL SIGMA LOCN
  1024 1645         474 RCR    8              C[2:0] = REL SIG REG LOCN
  1025 1646         406 A=C    X              A[2:0] = REL SIG REG LOCN
  1026 1647        1570 C=REGN 13             READ SIGMA, R00, .END. REG
  1027 1650          74 RCR    3              C[2:0] = REG 00 ADDRESS
  1028 1651         506 A=A+C  X              A[2:0] = NEW SIGMA REG ADR
  1029 1652         474 RCR    8              C[2:0] = OLD SIGMA REG ADR
  1030 1653         246 AC EX  X              C[2:0] = NEW SIGMA REG ADR
  1031 1654          74 RCR    3              ROTATE BACK TO NORMAL POSN
  1032 1655        1550 REGN=C 13             WRITE SIGMA/R00/.END. REG
  1033 1656        1740 RTN                   END OF READ STATUS FILE
*********************************************************************
* FNTDEV - TRY TO FIND ANOTHER DRIVE IN THE LOOP                     *
```

```
* OUTPUT:  RETURN TO P+1 IF ANOTHER DRIVE NOT FOUND                    *
*          RETURN TO P+2 IF ANOTHER DRIVE FOUND WITH                   *
*          R5R/W = NEXT DRIVE ADDRESS                                  *
* USES:    A, C, +2 SUBROUTINE LEVELS                                  *
*************************************************************************
  1041                   ENTRY  FNTDEV
*
  1043 1657 FNTDEV 1670 C=REGN 14            READ FLAGS REGISTER
  1044 1660         574 RCR    6             C[S] = FLAGS 32-35
  1045 1661         776 C=C+C  S             MANUAL MODE (FLAG 32) ?
  1046 1662        1540 RTN C                YES, DON'T SEARCH NEXT DRV
  1047 1663 SKPDEV    1 GOSUB  ASP           SEND AUTO CONFIGURE CMD
  1047 1664           0                      *ILCAS&CTL:  CS0, @0205
  1048 1665         674 RCR    11            C[M] = # OF DEVS IN LOOP
  1049 1666         432 A=C    M             A[M] = # OF DEVS IN LOOP
  1050 1667          36 A=0    S             CLEAR MANTISSA SIGN OF A
  1051 1670         576 A=A+1  S             A[S] = 1
  1052                   LEGAL                (CLEAR THE CARRY FLAG)
  1053 1671           1 GOSUB  FDEV20        GET # OF DEVICES IN LOOP
  1053 1672           0                      *ILCAS&CTL:  CS0, @0621
  1054 1673        1740 RTN                  RETURN IF DEV NOT FOUND
  1055 1674        1014 ?S2=1                NEW TAPE OR NO TAPE ?
  1056 1675          33 GONC   FNDNXT (1700) NO, FOUND NEXT DRIVE
  1057 1676        1414 ?S1=1                NO TAPE ?
  1058 1677        1643 GONC   SKPDEV (1663) YES, SKIP THIS DRIVE
  1059 1700 FNDNXT    1 GOLONG RTNP+2        RETURN TO CALL ADDR + 2
  1059 1701           2                      *ILCAS&CTL:  CS0, @0656
*
*************************************************************************
* INSTAT - READ A BYTE OF DEVICE STATUS                                 *
*          THE BYTE WILL BE STORED IN USER FLAG 0-7 (F0 IS LSB)         *
*          AND LOWER 6 BITS OF THE BYTE WILL BE CONVERTED INTO          *
*          A DECIMAL NUMBER AND STORED IN THE X-REGISTER.               *
*          IF MORE THAN ONE BYTE IS READ,                               *
*          ONLY STORE THE FIRST NON-ZERO BYTE.                          *
*          THE DEVICE HAS TO RESPOND TO RDY FRAME "SST",                *
*          OTHERWISE, IT WILL RETURN WITH ZERO.                         *
*************************************************************************
*
  1072                   ENTRY  INSTAT
*
  1074 1702         224 CON    @224          T
  1075 1703           1 CON    @01           A
  1076 1704          24 CON    @24           T
  1077 1705          23 CON    @23           S
  1078 1706          16 CON    @16           N
  1079 1707          11 CON    @11           I
  1080 1710 INSTAT    1 GOSUB  SCHDEV        GET DEVICE ADDRESS
  1080 1711           0                      *ILCAS&CTL:  CS0, @1335
  1081 1712           1 GOSUB  TALKER        MAKE IT A TALKER
  1081 1713           0                      *ILCAS&CTL:  CS0, @0262
  1082 1714         460 LDI                  SEND RDY FRAME - "SST"
  1083 1715         141 CON    @141          @141 = SEND STATUS
  1084 1716           1 GOSUB  RDTYPC        READ DEVICE TYPE CODE
  1084 1717           0                      *ILCAS&CTL:  CS0, @0466
  1085 1720         460 LDI                  LOAD MASK BITS
  1086 1721        1777 CON    @1777         TEN 1 BITS COMPRISE MASK
  1087 1722        1660 C=C.A                MASK OFF TOP STATUS BITS
  1088 1723         346 BC EX  X             B[X]=LOW 10 BITS FROM C
  1089 1724        1646 B SR   X             SAVE LOW 6 BITS IN B
```

```
  1090 1725           1 GOSUB  UNTCHK         SEND UNTALK, CHECK ERROR
  1090 1726           0                       *ILCAS&CTL:  CS1, @1745
  1091 1727        1134 PT=    9              REVERSE  ORDER OF 8 BITS
  1092 1730         246 AC EX  X              C[1:0]=REMAINING STAT BYTE
  1093 1731 STS10    26 A=0    XS             CLEAR BIT STORAGE AREA
  1094 1732         746 C=C+C  X              LEFT SHIFT 1 BIT TO C[XS]
  1095 1733          23 GONC   STS20  (1735)  TOP DIGIT IS ZERO
  1096 1734         566 A=A+1  XS             TOP DIGIT IS ONE
  1097 1735 STS20   246 AC EX  X              C[1:0]=REVERSED BITS AREA
  1098 1736         746 C=C+C  X              RIGHT SHIFT BIT TO C[1:0]
  1099 1737         746 C=C+C  X              SHIFT LEFT SECOND BIT
  1100 1740         746 C=C+C  X              SHIFT LEFT THIRD BIT
  1101 1741        1474 RCR    1              NOW ROTATE RIGHT 4 BITS
  1102 1742         246 AC EX  X              A=REVERSED BITS, C=STAT
  1103 1743        1724 DEC PT                DECREMENT BIT COUNTER
  1104 1744        1424 ? PT=  1              HAVE WE REACHED LOW BIT ?
  1105 1745        1643 GONC   STS10  (1731)  NO, KEEP READING BITS
  1106 1746        1670 C=REGN 14             YES, READ FLAGS REGISTER
  1107 1747        1574 RCR    12             C[1:0] = OLD FLAGS 0-7
  1108 1750         252 AC EX  WPT            C[1:0] = FLAGS FROM A[1:0]
  1109 1751        1074 RCR    2              C[13:12] = NEW FLAGS 0-7
  1110 1752        1650 REGN=C 14             REG.14[13:12] = NEW FLAGS
  1111 1753           1 GOSUB  ANNOUT         RESET DISPLAY ANNUNCIATORS
  1111 1754           0                       *MAINFRAME:  CN1, @1534
  1112 1755           1 GOLONG FNID60         PUT LOWER 6 BITS TO X-REG
  1112 1756           2                       *ILCAS&CTL:  CS3, @0062
  1113                  FILLTO @1760          START NEXT ROUTINE @1760
       1757        0000 NOP                   (INSERTED BY ASSEMBLER)
       1760        0000 NOP                   (INSERTED BY ASSEMBLER)
*
*********************************************************************
* RG-BY# - CONVERT NUMBER OF REGISTERS INTO NUMBER OF BYTES          *
*          (MULTIPLY THE REGISTER NUMBER BY 8)                       *
* INPUT:   C[3:0] = NUMBER OF REGISTERS                              *
* OUTPUT:  C[10:6]= NUMBER OF BYTES                                  *
*          C[5:2] = NUMBER OF REGS OR BYTES                          *
*          A[4:0] = NUMBER OF BYTES                                  *
*          PT = 4                                                    *
* USES:    A, C, PT, +0 SUBROUTINE LEVELS                            *
*********************************************************************
  1125                  ENTRY  RG-BY#
*
  1128 1761 RG-BY#  134 PT=    4             LOWEST MANTISSA BYTE
  1129 1762         102 C=0    PT            DIGIT 4 CLEARED
  1130 1763         412 A=C    WPT           A[4:0] = FILE SIZE IN REGS
  1131 1764         752 C=C+C  WPT           CONVERT # REGS TO # BYTES
  1132 1765         752 C=C+C  WPT             BY DOUBLING THREE TIMES,
  1133 1766         752 C=C+C  WPT             SAME AS MULTIPYING BY 8
  1134 1767         374 RCR    10            C[8:4] = NUMBER OF BYTES
  1135 1770          34 PT=    3             LOWEST MANTISSA DIGIT
  1136 1771         252 AC EX  WPT           C[3:0] = NUMBER OF REGS
  1137 1772         174 RCR    4             C[4:0] = NUMBER OF BYTES
  1138 1773         416 A=C                  A[4:0] = NUMBER OF BYTES
  1139 1774         474 RCR    8             C[10:6]=BYTES, C[5:2]=REGS
  1140 1775         134 PT=    4             LOWEST MANTISSA BYTE
  1141 1776        1740 RTN                   END OF RG-BY# SUBROUTINE
*
*
*
*
```

```
1146                    UNLIST
1149                    END

ERRORS :      0
```

SYMBOL TABLE (SCPL3B - ILCAS&CTL QUAD 2 = CS2 = ADDRESSES @74000-75777)

```
CKSMER  1246   -
CKSUME  1242   -
CLM10   1277   -
CLM15   1300   -  1307
CLNMEM  1271   -
CLTAIL  1310   -  1316
CX-AX    362   -   357
FLSCH     13   -
FLSCH0    12   -     7    4
FLSCHC    23   -   120
FLSCHD    10   -
FLSCHI     2   -
FLSCHJ     5   -
FLSCHX    16   -
FLSH01    31   -    26
FLSH05    32   -    30
FLSH10    53   -    75
FLSH15    70   -    60
FLSH20    76   -    67   45   43
FLSH30   101   -   114  105
FLSH32   103   -
FLSH35   121   -   117
FLSH70   160   -   153
FLSHDT   203   -
FLSHER   143   -
FLSHRT   173   -   161  151  147  124
FLTYCK   145   -   112   65
FLTYER   212   -   157
FNDNXT  1700   -  1675
FNTDEV  1657   -
INSTAT  1710   -
NOCST      0   -    20
NORMCK   705   -
NOROOM   716   -   712
RDENT    241   -
READP    620   -
READS   1565   -
READSB   611   -
RENT10   243   -
RENT15   246   -   260
RENT20   261   -   255
RENT30   263   -
RENT40   274   -   271
RENT45   304   -   301
RENT50   311   -   305  303
RG-BY#  1761   -
RP100    621   -   612
RP150    633   -   626
RP200    641   -   634
RP220    656   -   642
RP240    657   -   655  640
RP245    663   -   632
RP250    731   -   704
RP300    742   -   764
RP320    765   -   744
RP330   1000   -   751
```

```
RP335   1002   -   777
RP400   1026   -   774
RP401   1030   -
RP403   1043   -  1037
RP410   1064   -  1046
RP425   1103   -  1112
RP430   1104   -  1161
RP440   1152   -  1140
RP450   1155   -  1116 1114
RP460   1157   -  1154
RP470   1162   -  1130
RP480   1176   -  1173
RP490   1204   -  1175
RP500   1225   -  1222
RP550   1262   -  1235
RP560   1267   -  1263
RSTKCA  1320   -
RSTS35  1644   -  1633
RTKC10  1330   -  1365
RTKC15  1340   -  1367
RTKC20  1363   -  1355
RTKC30  1370   -  1347 1337
RTKC40  1404   -  1371
RTKC45  1406   -  1475 1417
RUSR25  1461   -  1450
RUSR30  1471   -  1446
RUSR40  1473   -  1470 1460 1441
RWCHK    225   -
RWCK20   232   -
SKPDEV  1663   -  1677
SKPF10   344   -   350
SKPF20   351   -   347
SKPF30   352   -   345
SKPFRM   341   -   361
STS10   1731   -  1745
STS20   1735   -  1733
WPNFER   420   -   424
WPROM    432   -   426
WRTP     402   -
WRTP00   403   -   375
WRTP10   430   -   414
WRTP15   435   -   431
WRTP20   437   -   427
WRTP30   534   -   531
WRTP40   537   -   535
WRTP70   554   -   570
WRTP80   573   -   556
WRTPV    374   -
WRTS    1502   -
```

ENTRY TABLE (SCPL3B - ILCAS&CTL QUAD 2 = CS2 = ADDRESSES @74000-75777)

CKSMER	1246	-
CKSUME	1242	-
CLM10	1277	-
CLNMEM	1271	-
CLTAIL	1310	-
CX-AX	362	-
FLSCH	13	-
FLSCH0	12	-
FLSCHC	23	-
FLSCHD	10	-
FLSCHI	2	-
FLSCHJ	5	-
FLSCHX	16	-
FLSHDT	203	-
FLSHRT	173	-
FLTYER	212	-
FNTDEV	1657	-
INSTAT	1710	-
NORMCK	705	-
NOROOM	716	-
RDENT	241	-
READP	620	-
READS	1565	-
READSB	611	-
RENT10	243	-
RG-BY#	1761	-
RP335	1002	-
RP400	1026	-
RP401	1030	-
RSTKCA	1320	-
RWCHK	225	-
SKPFRM	341	-
WPROM	432	-
WRTP	402	-
WRTPV	374	-
WRTS	1502	-

EXTERNAL REFERENCES (SCPL3B - ILCAS&CTL QUAD 2 = CS2 = ADR @74000-75777)

ANNOUT	1753					
ANNOUT	1754					
AOUT1	407					
AOUT1	410					
AOUTFL	35					
AOUTFL	36					
AOUTIN	14	443				
AOUTIN	15	444				
ASP	1663					
ASP	1664					
ASRCH	415					
ASRCH	416					
CKSMER	1242					
CKSMER	1243					
CKSUME	1615					
CKSUME	1616					
CLM10	1024					
CLM10	1025					
CLNMEM	1227					
CLNMEM	1230					
CLTAIL	1010	1165				
CLTAIL	1011	1166				
CNTBYT	511	700				
CNTBYT	512	701				
COPYBF	171					
COPYBF	172					
CPGM10	661					
CPGM10	662					
CPGMHD	470					
CPGMHD	471					
CRTFL	537					
CRTFL	540					
CRTFL0	1520					
CRTFL0	1521					
CSCKUT	600					
CSCKUT	601					
CSERCK	1240					
CSERCK	1241					
CSERE0	1244					
CSERE0	1245					
CSEREX	143	727				
CSEREX	144	730				
CSNOFD	0					
CSNOFD	1					
CSRDY	32					
CSRDY	33					
CX-AX	256					
CX-AX	257					
DDT0	241					
DDT0	242					
DECADA	1056	1060	1152	1163	1375	1377
DECADA	1057	1061	1153	1164	1376	1400
DSPERR	223	1642				
DSPERR	224	1643				
DUPFL	234					
DUPFL	235					

```
ENCP00  1320  1412
ENCP00  1321  1413
ERROR    432
ERROR    433
ERRPR    463
ERRPR    464
FDEV20  1671
FDEV20  1672
FLINKA   460
FLINKA   461
FLINKP   643
FLINKP   644
FLNMER   420
FLNMER   421
FLSCH   1503
FLSCH   1504
FLSCH0   623  1567
FLSCH0   624  1570
FLSCHX   446
FLSCHX   447
FNDCAS    16
FNDCAS    17
FNDEND  1526
FNDEND  1527
FNID60  1755
FNID60  1756
FNTDEV   115
FNTDEV   116
GCPKC0  1466
GCPKC0  1467
GENLNK  1133  1206
GENLNK  1134  1207
GETPC    435
GETPC    436
GTBYTA  1047  1123
GTBYTA  1050  1124
GTFEND   627   657  1404
GTFEND   630   660  1405
GTLINK  1407
GTLINK  1410
INCAD2   506  1121  1141  1425
INCAD2   507  1122  1142  1426
INCADA   647   760  1143  1310
INCADA   650   761  1144  1311
LDSST0   472   707
LDSST0   473   710
LJDLY   1260
LJDLY   1261
MEMLFT   664
MEMLFT   665
MESSLP  1246
MESSLP  1247
MSGROM   434
NATNRD   243
NATNRD   244
NFRPU   1267
NFRPU   1270
NRD      337   766
NRD      340   767
NRDC    1606
```

```
NRDC    1607
NXBYTA   560   651  1105  1427
NXBYTA   561   652  1106  1430
NXLDEL  1157
NXLDEL  1160
PACKE    714
PACKE    715
PILERR   571
PILERR   572
PLERCK   251
PLERCK   252
PLEREX   125   212   716  1634
PLEREX   126   213   717  1635
PTBYTA   762  1054  1146  1313  1454
PTBYTA   763  1055  1147  1314  1455
PURGEP   237
PURGEP   240
PUTPCX  1041
PUTPCX  1042
R5-R6     21
R5-R6     22
RDDFRM   246   330   746
RDDFRM   247   331   747
RDLPBK    50
RDLPBK    51
RDREG   1600
RDREG   1601
RDRGA   1602
RDRGA   1603
RDTYPC  1716
RDTYPC  1717
RENT10    54
RENT10    55
RENTPH   101
RENTPH   102
REGADR   162
REGADR   163
RG-BY#  1514
RG-BY#  1515
RSTKCA  1232
RSTKCA  1233
RTNP+2  1700
RTNP+2  1701
RWCHK    452  1507
RWCHK    453  1510
SCHDEV  1710
SCHDEV  1711
SDATA    565   576
SDATA    566   577
SEEKR2    76
SEEKR2    77
SEEKRD   167
SEEKRD   170
SEEKRN   731  1571
SEEKRN   732  1572
SEKSUB   541  1522
SEKSUB   542  1523
SETBP0    46
SETBP0    47
SIZSUB  1630
```

```
SIZSUB  1631
SKPFRM   265   274   312   314   317   321   324   326
SKPFRM   266   275   313   315   320   322   325   327
SNDATA   733
SNDATA   734
SRBMAP  1361  1471
SRBMAP  1362  1472
TALKER  1712
TALKER  1713
TBITMA  1357  1443
TBITMA  1360  1444
UNT      705  1026  1610
UNT      706  1027  1611
UNTCHK  1725
UNTCHK  1726
UPLINK  1414
UPLINK  1415
WKUP80  1265
WKUP80  1266
WRTA20  1556
WRTA20  1557

End of VASM assembly
********************************************************************************
********************************************************************************
********************************************************************************

VASM ROM ASSEMBLY          REV.  6/81A            HP-82160A HP-IL MODULE

OPTIONS: L C S          HP-IL CAS&CTL             ADDRESSES @76000-77777

     2                  FILE   SCPL4B             ILCAS&CTL QUAD 3 = CS3
*
*
********************************************************************************
* FINDID - GIVEN A DEVICE ID IN THE ALPHA REGISTER, RETURN                     *
*          WITH THE DEVICE LOOP ADDRESS IN THE X-REGISTER.                     *
*          X-REGISTER = 0, IF THE DEVICE ID NOT FOUND.                         *
*          THE SEARCH ALWAYS STARTS FROM DEVICE NUMBER 1.                      *
********************************************************************************
*
    12                  ENTRY  FINDID
    13                  ENTRY  FNID10        ** ADDED ON JUNE 4, 1981
*
    15    0          204 CON    @204          D
    16    1           11 CON    @11           I
    17    2            4 CON    @04           D
    18    3           16 CON    @16           N
    19    4           11 CON    @11           I
    20    5            6 CON    @06           F
    21    6 FINDID     1 GOSUB  SCHDEV        SEARCH FOR THE DEVICE
    21    7            0                       *ILCAS&CTL:  CS0, @1335
    22   10           10 S3=    1             SHIFT INTO M FROM LEFT END
    23   11            1 GOSUB  AOUT1         GET THE ID FROM A-REGISTER
    23   12            0                       *ILCAS&CTL:  CS0, @1064
    24   13 FNID10     1 GOSUB  TALKER        MAKE THE DEVICE A TALKER
    24   14            0                       *ILCAS&CTL:  CS0, @0262
    25   15            1 GOSUB  PLERCK        CHECK FOR ANY ERRORS
    25   16            0                       *ILCAS&CTL:  CS1, @1747
    26   17          144 HPL=CH 1             WRITE CONTROL INT REGISTER
```

```
    27   20         1205 CH=     @241           READY CLASS (BITS 7 & 5)
    28   21          244 HPL=CH  2              WRITE DATA BITS REGISTER
    29   22          611 CH=     @142           SEND "SDI" (SEND DEV ID)
    30   23          404 S8=     0              INCOMING DATA IS ALPHA
    31   24          116 C=0                    CLEAR ACCUMULATOR
    32   25          160 N=C                    CLEAR REGISTER N
    33   26         1004 S2=     0              DO NOT IGNORE CR/LF
    34   27            1 GOSUB   INADRD         READ THE DEVICE ID
    34   30            0                        *ILCAS&CTL:  CS0, @1577
    35   31         1434 PT=     1              POINT AT LOWEST BYTE
    36   32          260 C=N                    PUT RETURNED ID INTO C
    37   33         1356 ? C#0                  DID THE ID COME BACK 0 ?
    38   34          113 GONC    FNID35 (  45)  YES, TRY NEXT DEVICE ID
    39   35 FNID20 1352 ? C#0    WPT            IS THE ID IN LOWEST BYTE ?
    40   36           37 GOC     FNID30 (  41)  YES, COMPARE ID NUMBER
    41   37         1074 RCR     2              RIGHT-JUSTIFY THE ID IN C
    42   40         1753 GOTO    FNID20 (  35)  CHECK RIGHT-JUSTIFICATION
    43   41 FNID30  416 A=C                     RIGHT-JUSTIFIED ID INTO A
    44   42          630 C=M                    DESIRED DEVICE ID INTO C
    45   43         1556 ? A#C                  FOUND THE ID ?
    46   44           73 GONC    FNID45 (  53)  YES, GET DEVICE ADDRESS
    47   45 FNID35    1 GOSUB   PILEN          ENABLE PIL CHIP
    47   46           0                        *ILCAS&CTL:  CS0, @0143
    48   47           1 GOSUB   NXTDEV         GET NEXT DEVICE ADDRESS
    48   50           0                        *ILCAS&CTL:  CS0, @0541
    49   51          63 GOTO    FNID40 (  57)  P+1 - NO NEXT DEVICE
    50   52        1413 GOTO    FNID10 (  13)  P+2 - SEARCH NEXT DEVICE
    51   53 FNID45  544 C=HPIL  5              GET DEVICE ADDRESS
    51   54          572                       (INSERTED BY ASSEMBLER)
    51   55          503                       (INSERTED BY ASSEMBLER)
    52   56           53 GOTO    FNID65 (  63)  NORMALIZE BINARY TO BCD
*
    54   57 FNID40   56 B=0                     CLEAR REGISTER B
    55   60 FNIDRT    1 GOLONG  RCL            WRITE B TO REG X & EXIT
    55   61           2                        *MAINFRAME:  CN4, @1056
*
    57                ENTRY     FNID60
*
    59   62 FNID60  316 C=B                     GET DEVICE ADDRESS
    60   63 FNID65    1 GOSUB   BINBD0         NORMALIZE BINARY TO BCD
    60   64           0                        *ILCAS&CTL:  CS3, @1724
    61   65          36 A=0     S              CLEAR REG.A MANTISSA SIGN
    62   66         334 PT=     10             SET POINTER TO DIGIT 10
    63   67          12 A=0     WPT            CLEAR REGISTER A[10:0]
    64   70         106 C=0     X              CLEAR REGISTER C[2:0]
    65   71        1160 DADD=C                  SELECT CHIP 0
    66   72        1046 C=C+1   X              ASSUME GREATER THAN 9
    67   73         406 A=C     X              EXPONENT OF A IS 1
    68   74        1534 PT=     12             POINT AT DIGIT 12
    69   75        1502 ? A#0   PT             IS DIGIT 12 ZERO ?
    70   76          37 GOC     FNID70 ( 101)  NO, SAVE A TO B AND EXIT
    71   77        1772 A SL    M              YES, SHIFT A LEFT 1 DIGIT
    72  100           6 A=0     X              EXPONENT OF A NOW 0
    73  101 FNID70  156 AB EX                   SAVE REGISTER A TO B
    74  102        1563 GOTO    FNIDRT (  60)  WRITE B TO X AND EXIT
*
*******************************************************************
* STOPIO - SEND IFC (INTERFACE CLEAR)                             *
*******************************************************************
*
```

```
    80                  ENTRY  STOPIO
*
*
    83  103         217 CON    @217          O
    84  104          11 CON    @11           I
    85  105          20 CON    @20           P
    86  106          17 CON    @17           O
    87  107          24 CON    @24           T
    88  110          23 CON    @23           S
    89  111 STOPIO    1 GOSUB  SCHDEV        SEARCH FOR THE DEVICE
    89  112           0                       *ILCAS&CTL:  CS0, @1335
*
    91                  ENTRY  IFC
*
    93  113 IFC      44 HPL=CH 0              WRITE STATUS REGISTER
    94  114        1601 CH=    @340           SELF AS SC, CA, TA
    95  115         144 HPL=CH 1              WRITE CONTROL INT REGISTER
    96  116        1005 CH=    @201           COMMAND CLASS (BIT 7)
    97  117         244 HPL=CH 2              WRITE DATA BITS REGISTER
    98  120        1101 CH=    @220           SEND COMMAND - "IFC"
    99  121        1034 PT=    2              POINT AT EXPONENT SIGN
   100  122        1520 LC     13             TIME  OUT ONLY 1 SECOND
   101  123           1 GOSUB  SCMD20         WAIT FOR FRAME TO RETURN
   101  124           0                       *ILCAS&CTL:  CS0, @0276
   102  125          44 HPL=CH 0              WRITE STATUS REGISTER
   103  126        1611 CH=    @342           SET CLEAR IFC RECEIVED = 1
   104  127        1740 RTN                   END OF STOPIO & IFC
*
*************************************************************************
* PURGEF - PURGE A FILE (NON-PROGRAMMABLE) (USER FUNCTION IS "PURGE")  *
*          A FILE CAN'T BE PURGED IF IT IS PROTECTED                   *
*************************************************************************
*
   111                  ENTRY  PURGEF
   112                  ENTRY  PURGEP
   113                  ENTRY  REWENT
   114                  ENTRY  WRET10
   115                  ENTRY  WRET15
*
   117  130         205 CON    @205          E
   118  131           7 CON    @07           G
   119  132          22 CON    @22           R
   120  133          25 CON    @25           U
   121  134          20 CON    @20           P
   122  135 PURGEF    1 GOSUB  FLSCHJ        SEARCH THE FILE
   122  136           0                       *ILCAS&CTL:  CS2, @0005
   123  137          76 B=0    S              DO COPYBF
   124  140 PURGEP    1 GOSUB  CHKPCT        CHECK IF FILE IS PROTECTED
   124  141           0                       *ILCAS&CTL:  CS3, @0414
   125  142         260 C=N                   GET FILE INFORMATION
   126  143         136 C=0    S              C[S] = FILE TYPE
   127  144         160 N=C                   SAVE FILE INFORMATION
*************************************************************************
* REWENT - WRITE A FILE ENTRY TO DIRECTORY                              *
* ASSUMES: THE POINTER IS POINTING AT THE END OF A FILE ENTRY.          *
*          THIS ROUTINE WILL WRITE THE ENTRY OVER THE SAME PLACE.       *
* INPUT:   M = FILE NAME                                                *
*          N = FILE TYPE(2), STARTING RECORD NUMBER(4),                 *
*              FILE LENGTH IN RECORDS(4), FILE SIZE(4).                 *
*          IF B[S] = 0, WILL DO A COPY BUFFER 0 TO BUFFER 1 AT END.     *
```

```
* USES:    A, B[X], C, PT, S[7:0], +2 SUB LEVEL                          *
*********************************************************************
   138  145 REWENT    1 GOSUB  REQADR         READ CURRENT ADDRESS
   138  146           0                       *ILCAS&CTL:  CS3, @1100
   139  147         646 A=A-1  X              BACK UP 1 RECORD
   140  150        1312 ? B#0  WPT            POINTING AT REC BOUNDARY ?
   141  151          77 GOC    WRET10 ( 160) NO, SEEK RECORD DIRECTLY
   142  152         646 A=A-1  X              BACK UP ANOTHER RECORD
   143  153         206 B=A    X              SAVE REC # TO SEEK IN B
   144  154           1 GOSUB  SEEKRD         BACK UP 2 REC & READ IT
   144  155           0                       *ILCAS&CTL:  CS3, @1572
   145  156         146 AB EX  X              A[X] = RECORD NUMBER
   146  157          46 B=0    X              CLEAR B[X]
   147  160 WRET10    1 GOSUB  SEEK           SEEK TO A GIVEN RECORD
   147  161           0                       *ILCAS&CTL:  CS3, @1565
   148  162           1 GOSUB  SRWRT          SEND DDL2 COMMAND (WRITE)
   148  163           0                       *ILCAS&CTL:  CS0, @0323
   149  164         146 AB EX  X              A[1:0] = PTR OF AN ENTRY
   150  165         460 LDI                   C[1:0] = # OF BYTES
   151  166          40 CON    32             # OF BYTES TO SUBTRACT
   152  167         706 A=A-C  X              BACK UP 32 BYTES
   153  170           0 NOP                   CLEAR THE CARRY FLAG
   154  171 WRET15    1 GOSUB  SETBPT         SET BYTE POINTER
   154  172           0                       *ILCAS&CTL:  CS3, @1631
   155  173           1 GOSUB  DTFLOW         SEND SEC.CMD - DATA FOLLOW
   155  174           0                       *ILCAS&CTL:  CS0, @0332
   156  175         630 C=M                   READ FILE NAME TO REG C
   157  176         534 PT=    6              SET UP TO COUNT 6 TO 0
   158  177           1 GOSUB  SNBYTS         SEND 7 BYTES OF NAME
   158  200           0                       *ILCAS&CTL:  CS3, @1674
   159  201         260 C=N                   READ FILE INFO TO REG C
   160  202        1076 C=C+1  S              IS IT A 41C FILE ?
   161  203         717 GOC    WRET50 ( 274) NO, DON'T CHANGE TYPE
   162  204         260 C=N                   YES, C[S] = FILE TYPE
   163  205         416 A=C                   SAVE FILE INFO TO REG A
   164  206         132 C=0    M              CLEAR C MANTISSA DIGITS
   165  207         106 C=0    X              CLEAR C EXPONENT DIGITS
   166  210         474 RCR    8              C = 00000000T00000
   167  211        1334 PT=    13             POINT AT SIGN DIGIT
   168  212         220 LC     2              LOAD 3 BLANKS (HEX 20)
   169  213          20 LC     0
   170  214         220 LC     2
   171  215          20 LC     0
   172  216         220 LC     2              C = 20202000T00000
   173  217         676 A=A-1  S              IS IT A PURGED FILE ?
   174  220         117 GOC    WRET40 ( 231) YES, HANDLE PURGED FILE
   175  221         676 A=A-1  S              NO, IS IT AN ASCII FILE ?
   176  222          53 GONC   WRET30 ( 227) NO, SET C[7:6] = E0
   177  223         234 PT=    5              YES, SET C[5:4] = 01
   178  224          20 LC     0              CLEAR DIGIT 5
   179  225         120 LC     1              C = 20202000010000
   180  226          33 GOTO   WRET40 ( 231) WRITE LEFTMOST 5 BYTES
   181  227 WRET30 1234 PT=    7              POINT TO DIGIT 7
   182  230        1620 LC     14             C = 202020E0T00000
   183  231 WRET40  134 PT=    4              WRITE LEFTMOST 5 BYTES
   184  232           1 GOSUB  SNBYTS         SEND 2 BYTES OF TYPE
   184  233           0                       *ILCAS&CTL:  CS3, @1674
   185  234         260 C=N                   GET FILE INFO TO REG C
   186  235        1574 RCR    12             C[1:0] = FILE TYPE/PROTECT
   187  236          34 PT=    3              POINT AT LOWEST 2 BYTES
```

```
   188  237          112 C=0    WPT           CLEAR C[3:0] BEFORE ROTATE
   189  240          174 RCR    4             C[13:6] = 0000+START REC #
   190  241            1 GOSUB  SNBYTS        SEND 4 BYTES = START REC #
   190  242            0                      *ILCAS&CTL:  CS3, @1674
   191  243          260 C=N                  GET FILE INFO TO REG C
   192  244          474 RCR    8             C[13:10]=FILE LEN IN #RECS
   193  245           34 PT=    3             POINT AT LOWEST 2 BYTES
   194  246          112 C=0    WPT           CLEAR C[3:0] BEFORE ROTATE
   195  247          174 RCR    4             C[13:6] = 0000+FILE LENGTH
   196  250            1 GOSUB  SNBYTS        SEND 4 BYTES = FILE LENGTH
   196  251            0                      *ILCAS&CTL:  CS3, @1674
   197  252          116 C=0                  CLEAR ACCUMULATOR
   198  253          234 PT=    5             WRITE LEFTMOST 6 BYTES
   199  254            1 GOSUB  SNBYTS        SEND 6 BYTES = TIME (ZERO)
   199  255            0                      *ILCAS&CTL:  CS3, @1674
   200  256          116 C=0                  CLEAR ACCUMULATOR
   201  257         1334 PT=    13            POINT AT MANTISSA SIGN
   202  260         1020 LC     8             C[S] = 8
   203  261          334 PT=    10            POINT AT C[10]
   204  262          120 LC     1             C[13:10] = 8001 (VOL #)
   205  263         1434 PT=    1             WRITE LEFTMOST 2 BYTES
   206  264            1 GOSUB  SNBYTS        SEND 2 BYTES = VOLUME #
   206  265            0                      *ILCAS&CTL:  CS3, @1674
   207  266          260 C=N                  GET FILE INFO TO REG C
   208  267          136 C=0    S             CLEAR FILE TYPE DIGIT
   209  270          174 RCR    4             C[13:8]=FILE SIZE, OPTION
   210  271         1034 PT=    2             SEND 2 BYTES OF SIZE AND
   211  272            1 GOSUB  SNBYTS         1 BYTE OF FILE OPTION
   211  273            0                      *ILCAS&CTL:  CS3, @1674
   212  274 WRET50   144 HPL=CH 1             WRITE CONTROL INT REGISTER
   213  275          405 CH=    @101          END CLASS (BIT 6)
   214  276          460 LDI                  LOAD LOW 12 BITS OF C WITH
   215  277           40 CON    @40           @40 = ASCII BLANK
   216  300            1 GOSUB  SDATA         SEND END FRAME WITH BLANK
   216  301            0                      *ILCAS&CTL:  CS0, @0450
   217  302            1 GOSUB  WAITS         WAIT FOR DONE, CHECK ERROR
   217  303            0                      *ILCAS&CTL:  CS1, @0032
   218  304         1336 ? B#0  S             UPDATE DIRECTORY BUFFER ?
   219  305         1540 RTN C                NO, IF B[S] IS NON-ZERO
*
*****************************************************************
* COPYBF - COPY THE CURRENT DATA BUFFER TO DIRECTORY BUFFER.     *
*          FILBERT HAS TWO 256-BYTE BUFFERS, ONE FOR TRANSFERRING*
*          DATA TO/FROM TAPE (0), AND ONE FOR READ/WRITE LOOP BACK*
*          FUNCTION (1), THE LATTER OF WHICH IS NOT USED REGULARLY.*
*          THE SECOND BUFFER IS NOW USED TO KEEP THE LAST ACCESSED*
*          DIRECTORY RECORD, WHICH CAN SPEED UP THE FILE SEARCH IF*
*          THE NUMBER OF FILES ON THE TAPE IS SMALL.             *
*                                                                *
* USES:    A, C, +2 SUBROUTINE LEVELS                           *
* OUTPUT:  STORE A CONSTANT 250 TO BYTE #250 TO INDICATE LAST FILE*
*          ACCESSED WAS NOT A DATA FILE.                         *
*          LEAVES THE CASSETTE NOT A TALKER NOR A LISTENER.      *
*****************************************************************
   235                    ENTRY  COPYBF
*
   237  306 COPYBF     1 GOSUB  REQADR        READ CURRENT ADDRESS
   237  307            0                      *ILCAS&CTL:  CS3, @1100
   238  310            1 GOSUB  LISTEN        ADDRESS DEVICE AS LISTENER
   238  311            0                      *ILCAS&CTL:  CS0, @0335
```

```
   239  312          460 LDI                  DEV DEPENDENT LISTENER 9
   240  313          251 CON    @251          DDL9 = COPY BUFR1 TO BUFR2
   241  314            1 GOSUB  SCMD          SEND DDL9 TO CASSETTE
   241  315            0                      *ILCAS&CTL:  CS0, @0272
   242  316          460 LDI                  LOAD LOW 12 BITS OF C WITH
   243  317          372 CON    250           BYTE POINTER VALUE OF 250
   244  320            1 GOSUB  SETBPC        SET BYTE POINTER TO 250
   244  321            0                      *ILCAS&CTL:  CS3, @1645
   245  322            1 GOSUB  WRLPBK        GOTO WRITE LOOP BACK MODE
   245  323            0                      *ILCAS&CTL:  CS1, @1660
   246  324          460 LDI                  LOAD LOW 12 BITS OF C WITH
   247  325          372 CON    250           BYTE VALUE OF 250
   248  326            1 GOSUB  SDATA0        SEND A BYTE OF DATA
   248  327            0                      *ILCAS&CTL:  CS0, @0446
   249  330          146 AB EX  X              A[X] = NUMBER OF BYTES
   250  331            1 GOSUB  SETBPL        SET THE BYTE POINTER BACK
   250  332            0                      *ILCAS&CTL:  CS3, @1627
   251  333            1 GOLONG UNL           SEND UNLISTEN COMMAND
   251  334            2                      *ILCAS&CTL:  CS0, @0257
*************************************************************************
* CHKCST - LOOK FOR THE CASSETTE IN THE LOOP AND THEN READ              *
*          ITS STATUS TO SEE IF IT IS IN IDLE.                          *
*          IF CASSETTE NOT FOUND OR CASSETTE IS BUSY IT WILL BE         *
*          DIRECTED TO AN ERROR EXIT IN THE COCONUT MAINFRAME AND       *
*          WILL NOT RETURN TO THE CALLING PROGRAM.                      *
* ASSUMES: NOTHING                                                      *
* OUTPUT:  CASSETTE STATUS IN S[7:0]                                    *
*          CASSETTE ADDRESS IN R5R/W                                    *
* USES:    A, C, S[7:0], NO PT, +2 SUBROUTINE LEVELS                    *
*************************************************************************
   263                   ENTRY  CHKCST        CHECK CASS PRESENT & READY
   264                   ENTRY  CSNOFD        CASSETTE NOT FOUND
   265                   ENTRY  CSEREX        CASSETTE ERROR EXIT
   266                   ENTRY  CSERE0
   267                   ENTRY  CSRDY         CHECK IF CASSETTE READY
   268                   ENTRY  INTDIR
   269                   ENTRY  CHKCS0
*
   271  335 CHKCST     404 S8=    0           S8=0: CALLED FROM CHKCST
   272  336 CHKCS0       1 GOSUB  FNDCAS       LOOK FOR THE CASSETTE
   272  337              0                     *ILCAS&CTL:  CS0, @0566
   273  340            243 GOTO   CSNOFD ( 364) P+1 - CASSETTE NOT FOUND
   274  341 CSRDY      214 ?S5=1                P+2 - CASSETTE BUSY ?
   275  342             63 GONC   CSRDY1 ( 350) NO, CASSETTE IS READY
   276  343              1 GOSUB  CSSTAS       YES, READ CASSETTE STATUS
   276  344              0                     *ILCAS&CTL:  CS0, @0700
   277  345              1 GOSUB  PLERCK       CHECK FOR ANY ERRORS
   277  346              0                     *ILCAS&CTL:  CS1, @1747
   278  347           1723 GOTO   CSRDY  ( 341) CHECK FOR CASSETTE READY
   279  350 CSRDY1     414 ?S8=1                CHKCS0 CALLED FROM NEWM?
   280  351           1540 RTN C               YES, RETURN TO NEWM
   281  352            460 LDI                 LOAD LOW 12 BITS OF C WITH
   282  353              7 CON    7            BYTE CONTAINING VALUE OF 7
   283  354            406 A=C    X            A[X] = 7 (NEW TAPE STATUS)
   284  355           1634 PT=    0            POINT AT LOWEST DIGIT
   285  356           1630 C=ST                COPY STATUS TO LOWEST BYTE
   286  357           1542 ? A#C  PT           IS THIS A NEW TAPE ?
   287  360           1540 RTN C               YES, RETURN TO CALLER
   288  361 INTDIR       1 GOSUB  SEEKR2       RESTORE DIR BUFR F/REC 0
   288  362              0                     *ILCAS&CTL:  CS3, @1557
```

```
   289  363         1233 GOTO   COPYBF ( 306) COPY DATA TO DIR BUFFER
   290  364 CSNOFD     1 GOSUB  PLEREX         DISPLAY "NO DRIVE"
   290  365             0                     *ILCAS&CTL:  CS1, @1663
   291  366            16 CON    @16           N
   292  367            17 CON    @17           O
   293  370            40 CON    @40                BLANK
   294  371             4 CON    @04           D
   295  372            22 CON    @22           R
   296  373            11 CON    @11           I
   297  374            26 CON    @26           V
   298  375          1005 CON    @1005         E
   299  376 CSEREX     1 GOSUB  LEFTJ          LEFT-JUSTIFY DISPLAY
   299  377             0                     *MAINFRAME: CN10, @1767
   300  400 CSERE0   410 S8=    1             PRINT & SET MESSAGE FLAG
   301  401             1 GOSUB  MSG105        PRINT DISP IN NORM & TRACE
   301  402             0                     *MAINFRAME:  CN7, @0200
   302  403             1 GOLONG ERR110        ERROR EXIT W/WO BACKUP
   302  404             2                     *MAINFRAME:  CN8, @1373
*
   304                   ENTRY  DSPERR         DISPLAY MESSAGE: " ERR"
*
   306  405 DSPERR     1 GOSUB  MESSL          DISPLAY MESSAGE: ERR
   306  406             0                     *MAINFRAME:  CN1, @1757
   307  407            40 CON    @40                BLANK
   308  410             5 CON    @05           E
   309  411            22 CON    @22           R
   310  412          1022 CON    @1022         R
   311  413          1633 GOTO   CSEREX ( 376) CASSETTE ERROR EXIT
*************************************************************************
* CHKPCT - CHECK IF THE FILE IS PROTECTED                               *
*          IN EVERY FILE ENTRY THERE ARE TWO BITS IN THE FILE TYPE      *
*          USED FOR FILE PROTECTION. THE MSB (BIT 7) OF THE FILE TYPE   *
*          IS THE USER PROTECT BIT WHICH CAN BE SET OR RESET BY THE     *
*          USER THROUGH THE TWO PROTECT FUNCTIONS.                      *
*************************************************************************
   319                   ENTRY  CHKPCT         CHECK: IS FILE PROTECTED ?
*
   321  414 CHKPCT   260 C=N                   CHECK IF FILE IS PROTECTED
   322  415          1374 RCR    13            ROTATE LEFT ONE DIGIT
   323  416           776 C=C+C  S             IS PROTECTION BIT SET ?
   324  417          1640 RTN NC               NO, RETURN IMMEDIATELY
   325  420 FLPT10     1 GOSUB  PLEREX         PIL ERROR EXIT: FL SECURED
   325  421             0                     *ILCAS&CTL:  CS1, @1663
   326  422             6 CON    @06           F
   327  423            14 CON    @14           L
   328  424            40 CON    @40                BLANK
   329  425            23 CON    @23           S
   330  426             5 CON    @05           E
   331  427             3 CON    @03           C
   332  430            25 CON    @25           U
   333  431            22 CON    @22           R
   334  432             5 CON    @05           E
   335  433          1004 CON    @1004         D
   336  434          1423 GOTO   CSEREX ( 376) CASSETTE ERROR EXIT
*
*************************************************************************
* VERIFY - VERIFY A FILE                                                *
*          WILL ONLY READ THE FILE INTO DRIVER'S BUFFER, NOT TRANSMIT   *
*          THE DATA OUT, BECAUSE ANY PIL TRANSMIT ERROR SHOULD BE       *
*          DETECTED WHILE GENERATING THE FILE.  THE ONLY THING THAT     *
```

```
*          MIGHT GO WRONG IS THAT THE DRIVER MAY NOT RECORD ON THE     *
*          TAPE CORRECTLY.                                             *
*************************************************************************
*
   347                  ENTRY  VERIFY
   348  435          231 CON    @231          Y
   349  436            6 CON    @06           F
   350  437           11 CON    @11           I
   351  440           22 CON    @22           R
   352  441            5 CON    @05           E
   353  442           26 CON    @26           V
*
   355  443 VERIFY     1 GOSUB  FLSCHI        READ THE FILE ENTRY
   355  444            0                      *ILCAS&CTL:  CS2, @0002
   356  445            1 GOSUB  SEEKRN        SEEK & READ 1ST RECORD
   356  446            0                      *ILCAS&CTL:  CS3, @1567
   357  447          260 C=N                  GET FILE INFO FROM REC N
   358  450          174 RCR    4             C[X] = # OF RECS IN FILE
   359  451         1146 C=C-1  X             DECREMENT BY 1 RECORD
   360  452          346 BC EX  X             B[X] = # OF RECORDS - 1
   361  453 VERF10   146 AB EX  X             A[X] = RECORD COUNT
   362  454          646 A=A-1  X             ALL DONE ?
   363  455            1 GOLC   UNT           SEND UNTALK COMMAND
   363  456            3                      *ILCAS&CTL:  CS0, @0254
   364  457          206 B=A    X             SAVE RECORD COUNT TO B[X]
   365  460            1 GOSUB  SEEK40        READ ONE RECORD
   365  461            0                      *ILCAS&CTL:  CS0, @1640
   366  462         1713 GOTO   VERF10 ( 453) GET NEXT RECORD
*
*************************************************************************
* WRTA - WRITE ALL                                                      *
*************************************************************************
*
   372                  ENTRY  WRTA
   373                  ENTRY  WRTA20
*
   375  463          201 CON    @201          A
   376  464           24 CON    @24           T
   377  465           22 CON    @22           R
   378  466           27 CON    @27           W
   379  467 WRTA     314 ?S10=1               ARE WE IN ROM ?
   380  470            1 GOLC   WPROM         NOT ALLOWED WHILE IN ROM
   380  471            3                      *ILCAS&CTL:  CS2, @0432
   381  472         1770 C=REGN 15            C[X] = CURRENT LINE NUMBER
   382  473          106 C=0    X             CLEAR CURRENT LINE NUMBER
   383  474         1146 C=C-1  X             CURRENT LINE NUMBER = FFF
   384  475         1750 REGN=C 15            WRITE CURRENT LINE NUMBER
   385  476            1 GOSUB  GTFEND        GET FINAL END ADDRESS
   385  477            0                      *MAINFRAME:  CN8, @0350
   386  500 WRTA10     1 GOSUB  FLINKA        FIND PROGRAM END ADDRESS
   386  501            0                      *MAINFRAME: CN10, @0447
   387  502         1514 ?S12=1               IS IT A PRIVATE PROGRAM ?
   388  503            1 GOLC   ERRPR         YES, ERROR: PRIVATE
   388  504            3                      *MAINFRAME:  CN8, @0604
   389  505         1506 ? A#0  X             REACHED REGISTER 0 YET ?
   390  506         1727 GOC    WRTA10 ( 500) NOT YET, KEEP LOOKING
   391  507          116 C=0                  CLEAR ACCUMULATOR
   392  510         1160 DADD=C               ENABLE CHIP 0
   393  511            1 GOSUB  FLSCH         CHECK DUPLICATION
   393  512            0                      *ILCAS&CTL:  CS2, @0013
```

```
   394  513          460 LDI                    LOAD LOW 12 BITS OF C WITH
   395  514            4 CON    4               WRITE ALL FILE TYPE
   396  515            1 GOSUB  RWCHK           CHECK FOR OVERWRITE
   396  516            0                        *ILCAS&CTL:  CS2, @0225
   397  517            1 GOSUB  FNDEND          FIND TOP END OF MEMORY
   397  520            0                        *MAINFRAME:  CN5, @1460
   398  521          116 C=0                    CLEAR ACCUMULATOR
   399  522         1160 DADD=C                 ENABLE CHIP 0
   400  523         1670 C=REGN 14              READ FLAGS REGISTER
   401  524          674 RCR    11              C[1:0] = FLAGS 4-11
   402  525         1530 ST=C                   S0 = USER FLAG 11
   403  526          116 C=0                    CLEAR ACCUMULATOR
   404  527          460 LDI                    LOAD LOW 12 BITS OF C WITH
   405  530          257 CON2   10     15       AF = 176 REGISTER GAP - 1
   406  531         1106 C=A-C  X               C[X] = TOTAL # OF REGS
   407                   LEGAL                  (CLEAR THE CARRY FLAG)
   408  532            1 GOSUB  RG-BY#          COMPUTE NUMBER OF REGS
   408  533            0                        *ILCAS&CTL:  CS2, @1761
   409  534         1074 RCR    2               C[3:0] = # OF REGISTERS
   410  535         1152 C=C-1  WPT             DECREMENT ONE REGISTER
   411  536         1574 RCR    12              C[1:0] = FILE TYPE/PROTECT
   412  537         1614 ?S0=1                  WANT TO SET UP AUTO RUN ?
   413  540           33 GONC   WRTA15 ( 543)   NO, NORMAL RUNNING
   414  541         1634 PT=    0               POINT TO PROTECTION DIGIT
   415  542          220 LC     2               C[0] = 2 = AUTO RUN
   416  543 WRTA15  1434 PT=    1               POINT TO FILE TYPE DIGIT
   417  544          420 LC     4               FILE TYPE 4 = WRITE ALL
   418  545            1 GOSUB  CRTFL           CREATE THE FILE
   418  546            0                        *ILCAS&CTL:  CS1, @1260
   419  547            1 GOSUB  SEKSUB          SEEK TO BEGINNING OF FILE
   419  550            0                        *ILCAS&CTL:  CS0, @0326
   420  551          116 C=0                    SET UP TO WRITE CHIP 0
   421  552          460 LDI                    LOAD LOW 12 BITS OF C WITH
   422  553           20 CON    16              16 REGISTERS TO WRITE
   423  554          530 M=C                    SAVE # REGS TO WRITE IN M
   424  555           56 B=0                    INITIALIZE CHECKSUM
   425  556          410 S8=    1               S8=1: CALLED BY WRITEALL
   426  557            1 GOSUB  SNDRGA          SEND CHIP 0 (16 REGISTERS)
   426  560            0                        *ILCAS&CTL:  CS1, @0623
   427  561          260 C=N                    C[3:0] = TOTAL # OF REGS
   428  562           74 RCR    3               C[13:11] = TOTAL # OF REGS
   429  563          460 LDI                    LOAD LOW 12 BITS OF C WITH
   430  564          300 CON2   12     0        C0 = REGISTER 00 ADDRESS
   431  565         1574 RCR    12              C[1:0] = # OF REGS / 16
   432  566         1146 C=C-1  X               SUBTRACT 16 REGISTERS
   433  567         1374 RCR    13              C[2:0] = # OF REGS - 16
   434  570          530 M=C                    M[4:3]=C0, M[2:0]=# REGS
   435  571 WRTA20   410 S8=    1               S8=1: CALLED BY WRITEALL
   436  572            1 GOSUB  SNDRGA          SEND REGS STARTING AT C0
   436  573            0                        *ILCAS&CTL:  CS1, @0623
   437  574          316 C=B                    CHECKSUM TO REGISTER C
   438  575            1 GOSUB  SNDRGC          WRITE THE CHECKSUM
   438  576            0                        *ILCAS&CTL:  CS1, @0717
   439  577            1 GOLONG SNDRDN          CLOSE THE FILE
   439  600            2                        *ILCAS&CTL:  CS1, @0650
*
*****************************************************************
* READA - READ ALL                                              *
*****************************************************************
*
```

```
   445                  ENTRY  READA
   446                  ENTRY  COLDER
*
   448  601          201 CON    @201          A
   449  602            4 CON    @04           D
   450  603            1 CON    @01           A
   451  604            5 CON    @05           E
   452  605           22 CON    @22           R
   453  606 READA    460 LDI                  LOAD LOW 12 BITS OF C WITH
   454  607            4 CON    4             WRITE ALL FILE TYPE
   455  610            1 GOSUB  FLSCH0        SEARCH THE WRITE-ALL FILE
   455  611            0                      *ILCAS&CTL:  CS2, @0012
   456  612            1 GOSUB  FNDEND        FIND ADDR OF TOP OF MEMORY
   456  613            0                      *MAINFRAME:  CN5, @1460
   457  614          460 LDI                  LOAD LOW 12 BITS OF C WITH
   458  615          260 CON2   11     0      B0 = 176 NONEXISTENT REGS
   459  616          706 A=A-C  X             A[X] = TOTAL # REGS IN 41C
   460  617          260 C=N                  GET FILE SIZE
   461  620         1406 ? A<C  X             ENOUGH ROOM TO READ IT IN?
   462  621            1 GOLC   NOROOM        NO, DISPLAY "NO ROOM"
   462  622            3                      *ILCAS&CTL:  CS2, @0716
   463  623            1 GOSUB  SEEKRN        SEEK TO THE FILE
   463  624            0                      *ILCAS&CTL:  CS3, @1567
   464  625          116 C=0                  CLEAR ACCUMULATOR
   465  626          460 LDI                  LOAD LOW 12 BITS OF C WITH
   466  627           20 CON    16            READ 16 REGS OF CHIP 0
   467  630          530 M=C                  M[1:0] = 16 REGS TO READ
   468  631           56 B=0                  INITIALIZE CHECKSUM
   469  632            1 GOSUB  RDREG         READ FIRST 16 REGISTERS
   469  633            0                      *ILCAS&CTL:  CS1, @1020
   470  634          260 C=N                  C[3:0] = FILE SIZE
   471  635           74 RCR    3             C[13:11] = FILE SIZE
   472  636          460 LDI                  LOAD LOW 12 BITS OF C WITH
   473  637          300 CON2   12     0      C0 = REGISTER 00 ADDRESS
   474  640         1574 RCR    12            C[1:0] = FILE SIZE / 16
   475  641         1146 C=C-1  X             SUBTRACT 16 REGISTERS
   476  642         1374 RCR    13            C[2:0] = FILE SZ - 16 REGS
   477  643            1 GOSUB  RDRG10        READ REGISTERS FROM C0
   477  644            0                      *ILCAS&CTL:  CS1, @1023
   478  645            1 GOSUB  RDRGA         READ CHECKSUM
   478  646            0                      *ILCAS&CTL:  CS1, @1076
   479  647          160 N=C                  SAVE CHECKSUM IN N
   480  650          674 RCR    11            ROTATE BACK AFTER RDRGA
   481  651            1 GOSUB  NRDC          SEND "NRD" COMMAND
   481  652            0                      *ILCAS&CTL:  CS0, @0366
   482  653            1 GOSUB  UNT           SEND UNTALK COMMAND
   482  654            0                      *ILCAS&CTL:  CS0, @0254
   483  655          260 C=N                  GET CHECKSUM FROM FILE
   484  656          156 AB EX                GET CALCULATED CHECKSUM
   485  657         1556 ? A#C                DO CHECKSUMS MATCH ?
   486  660           53 GONC   RALL10 ( 665) YES, CHECK FLAGS & EXIT
   487  661 COLDER    1 GOSUB  CKSMER         NO, DISPLAY "READ ERR"
   487  662            0                      *ILCAS&CTL:  CS2, @1246
   488  663            1 GOLONG COLDST        DO COLD START
   488  664            2                      *MAINFRAME:  CN0, @1062
   489  665 RALL10  1504 S12=   0             CLEAR PRIVATE FLAG
   490  666          304 S10=   0             CLEAR ROM FLAG
   491  667            1 GOSUB  LDSST0        READ FLAGS REGISTER
   491  670            0                      *MAINFRAME:  CN1, @1627
   492  671          674 RCR    11            C[1:0] = FLAGS 4-11
```

```
   493  672         1530 ST=C                 S0 = USER FLAG 11
   494  673         1614 ?S0=1                WAS USER FLAG 11 SET ?
   495  674            1 GOLC   WKUP80        YES, SET OFF TO RUN
   495  675            3                      *MAINFRAME:  CN0, @0777
   496  676            1 GOLONG NFRC          NORMAL FUNCTION RETURN
   496  677            2                      *MAINFRAME:  CN0, @0361
*
*************************************************************************
* WRTK - WRITE KEY ASSIGNMENTS TO A "KEY" FILE                          *
*************************************************************************
*
   502                   ENTRY  WRTK
*
   504  700          213 CON    @213          K
   505  701           24 CON    @24           T
   506  702           22 CON    @22           R
   507  703           27 CON    @27           W
   508  704 WRTK     136 C=0    S             C[S]=0 CHECK FOR DUPLICATE
   509  705            1 GOSUB  FLSCH         CHECK FILE DUPLICATION
   509  706            0                      *ILCAS&CTL:  CS2, @0013
   510  707          460 LDI                  LOAD LOW 12 BITS OF C WITH
   511  710            5 CON    5             TYPE 5 = 41C KEY FILE
   512  711            1 GOSUB  RWCHK         CHECK FOR OVERWRITE
   512  712            0                      *ILCAS&CTL:  CS2, @0225
   513  713         1570 C=REGN 13            C[X] = LOCATION OF .END.
   514  714          346 BC EX  X             B[X] = CHAIN HEAD (.END.)
   515  715          460 LDI                  LOAD LOW 12 BITS OF C WITH
   516  716          277 CON2   11     15     SEE HOW MANY KEY REGISTERS
   517  717 WRTK10  1046 C=C+1  X             INCREMENT REGISTER NUMBER
   518  720         1160 DADD=C               POINT TO NEW REGISTER
   519  721          246 AC EX  X             SAVE POINTER TO REGISTER A
   520  722         1446 ? A<B  X             REACH CHAIN HEAD ?
   521  723           63 GONC   WRTK15 ( 731) YES, GET OUT OF LOOP
   522  724           70 C=DATA               RETRIEVE REGISTER CONTENTS
   523  725          246 AC EX  X             MOVE POINTER TO REGISTER C
   524  726         1076 C=C+1  S             IS THIS A KEY REGISTER ?
   525  727         1707 GOC    WRTK10 ( 717) YES, INCREMENT REG NUMBER
   526  730          406 A=C    X             SAVE POINTER TO REGISTER A
   527  731 WRTK15   460 LDI                  LOAD LOW 12 BITS OF C WITH
   528  732          300 CON2   12     0      C0 = REGISTER 00 ADDRESS
   529  733          706 A=A-C  X             A[X] = NUMBER OF KEY REGS
   530  734         1506 ? A#0  X             NUMBER OF KEY REGS = 0 ?
   531  735          243 GONC   WRTK20 ( 761) YES, EXIT W/ERROR: NO KEYS
   532  736          246 AC EX  X             MOVE # OF KEY REGS TO C
   533  737          132 C=0    M             CLEAR REST OF REGISTER C
   534  740            1 GOSUB  RG-BY#        CONVERT TO # OF BYTES
   534  741            0                      *ILCAS&CTL:  CS2, @1761
   535  742         1434 PT=    1             POINT TO LOWEST BYTE
   536  743          520 LC     5             TYPE 5 = 41C KEY FILE
   537  744            1 GOSUB  CRTFL0        CREATE THE KEY FILE
   537  745            0                      *ILCAS&CTL:  CS1, @1256
   538  746            1 GOSUB  SEKSUB        SEEK TO THE FILE
   538  747            0                      *ILCAS&CTL:  CS0, @0326
   539  750          260 C=N                  C[X] = # OF KEY REGISTERS
   540  751           74 RCR    3             C[13:11] = # OF KEY REGS
   541  752          460 LDI                  LOAD LOW 12 BITS OF C WITH
   542  753          300 CON2   12     0      C0 = START WRITING ADDRESS
   543  754          674 RCR    11            C[X] = # OF KEY REGISTERS
   544  755          530 M=C                  START REG AND # REGS TO M
   545  756           56 B=0                  INITIALIZE CHECKSUM
```

```
   546  757           1 GOLONG WRTA20        SEND REGS STARTING AT C0
   546  760           2                      *ILCAS&CTL:  CS3, @0571
   547  761 WRTK20    1 GOSUB  PLEREX        PIL ERR EXIT: NO KEYS
   547  762           0                      *ILCAS&CTL:  CS1, @1663
   548  763          16 CON    @16           N
   549  764          17 CON    @17           O
   550  765          40 CON    @40                BLANK
   551  766          13 CON    @13           K
   552  767           5 CON    @05           E
   553  770          31 CON    @31           Y
   554  771        1023 CON    @1023         S
   555  772           1 GOLONG CSEREX        CASSETTE ERROR EXIT
   555  773           2                      *ILCAS&CTL:  CS3, @0376
*
*****************************************************************
* READK - READ KEY ASSIGNMENTS FROM A "KEY" FILE                *
*****************************************************************
*
   561                 ENTRY  READK
*
   563  774         213 CON    @213          K
   564  775           4 CON    @04           D
   565  776           1 CON    @01           A
   566  777           5 CON    @05           E
   567 1000          22 CON    @22           R
   568 1001 READK   460 LDI                   LOAD LOW 12 BITS OF C WITH
   569 1002           5 CON    5             TYPE 5 = 41C KEY FILE
   570 1003           1 GOSUB  FLSCH0        SEARCH THE "KEY" FILE
   570 1004           0                      *ILCAS&CTL:  CS2, @0012
   571 1005           1 GOSUB  SEEKRN        SEEK AND READ THE RECORD
   571 1006           0                      *ILCAS&CTL:  CS3, @1567
   572 1007        1570 C=REGN 13            GET CHAIN HEAD
   573 1010         346 BC EX  X             SAVE IT IN B[X]
   574 1011           1 GOSUB  MEMLFT        COMPUTE # OF ZERO REGS
   574 1012           0                      *MAINFRAME:  CN1, @0641
   575 1013         146 AB EX  X             A[X]=ADDR OF CHAIN HEAD
   576 1014         706 A=A-C  X             A[X]=ADDR OF LAST ZERO REC
   577 1015         260 C=N                  FILE INFORMATION TO REG C
   578 1016         132 C=0    M             CLEAR OUT C[12:3]
   579 1017         674 RCR    11            C[13:11] = FILE SIZE BYTES
   580 1020         272 AC EX  M             SAVE FILE SIZE INTO TO A
   581 1021         260 C=N                  FILE INFORMATION TO REG C
   582 1022         174 RCR    4             SAVE FILE SIZE IN N[13:10]
   583 1023         460 LDI                  LOAD LOW 12 BITS OF C WITH
   584 1024         300 CON2   12     0      C0 = REGISTER 00 ADDRESS
   585 1025 RDKY10 1160 DADD=C               ENABLE NEXT KEY REGISTER
   586 1026         160 N=C                  SAVE ADDR OF CURR KEY REG
   587 1027          70 C=DATA               GET NEXT REGISTER CONTENTS
   588 1030        1076 C=C+1  S             IS THIS A KEY REGISTER ?
   589 1031         133 GONC   RDKY40 (1044) NO, MAY BE DONE OR LIFT IT
   590 1032 RDKY15 1532 ? A#0  M             DONE WITH REGS IN FILE ?
   591 1033          53 GONC   RDKY30 (1040) YES, REPL BY BLANK KEY REG
   592 1034         672 A=A-1  M             NO, DEC. # OF REGS IN FILE
   593 1035 RDKY20  260 C=N                  GET ADDR OF CURR KEY REG
   594 1036        1046 C=C+1  X             POINT TO NEXT KEY REGISTER
   595                 LEGAL                 (CLEAR THE CARRY FLAG)
   596 1037        1663 GOTO   RDKY10 (1025) ENABLE NEXT KEY REGISTER
* IF EXISTING NUMBER OF KEY REGISTERS > REGISTERS IN FILE,
* REPLACE A BLANK KEY REGISTER IN THOSE EXTRA OLD KEY REGS
   599 1040 RDKY30  116 C=0                  CLEAR THE ACCUMULATOR
```

```
   600 1041         1176 C=C-1  S             C[S] = F (HEX)
   601 1042         1360 DATA=C               WRITE 1 EMPTY KEY REGISTER
   602 1043         1723 GOTO   RDKY20 (1035) LOOK AT NEXT KEY REG
   603 1044 RDKY40  1532 ? A#0  M             DONE WITH REGS IN FILE ?
   604 1045          163 GONC   RDKY50 (1063) YES, READY TO READ
* IF EXISTING KEY REGS < REGS IN FILE, PRECREATE BLANK KEY REGISTER BY
* LIFTING UP ONE REGISTER FOR EACH ADDITIONAL KEY REGISTER
   607 1046          116 C=0                  CLEAR ACCUMULATOR
   608 1047         1160 DADD=C               ENABLE CHIP 0
   609 1050         1176 C=C-1  S             C[S] = F (HEX)
   610 1051          356 BC EX                CLEAR B[X]
   611 1052         1570 C=REGN 13            GET CHAIN HEAD (.END.)
   612 1053         1546 ? A#C  X             LAST ZERO REG=CHAIN HEAD ?
   613 1054            1 GOLNC  NORMCK        YES, "NO ROOM"/"TRY AGAIN"
   613 1055            2                      *ILCAS&CTL:  CS2, @0705
   614 1056          260 C=N                  GET ADDR OF CURR KEY REG
   615 1057            1 GOSUB  ASN15         LIFT UP ONE REGISTER
   615 1060            0                      *MAINFRAME:  CN9, @1702
   616 1061          546 A=A+1  X             INC. ADDR OF LAST ZERO REG
   617                   LEGAL                (CLEAR THE CARRY FLAG)
   618 1062         1503 GOTO   RDKY15 (1032) DEC REGS, GOTO NXT KEY REG
*
   620 1063 RDKY50   260 C=N                  SET M[X]=# OF REGS TO READ
   621 1064          374 RCR    10                 M[M]=START LOADING ADR
   622 1065          132 C=0    M             C[12:3]=0, C[X]=# REGS
   623 1066          134 PT=    4             POINT AT DIGIT 4
   624 1067         1420 LC     12            C[5:3]=C0 (START LOADING)
   625 1070          530 M=C                  SAVE TO M AS SHOWN ABOVE
   626 1071            1 GOSUB  RDREG0        READ REGISTERS ROUTINE
   626 1072            0                      *ILCAS&CTL:  CS1, @1017
   627 1073         1010 S2=    1             CALLED FROM READK
   628 1074            1 GOSUB  RSTKCA        REBUILD KEY ASSIGNMENTS
   628 1075            0                      *ILCAS&CTL:  CS2, @1320
   629 1076 RDKYER     1 GOLONG PLERCK        CHECK FOR ANY ERRORS
   629 1077            2                      *ILCAS&CTL:  CS1, @1747
*
*************************************************************************
* REQADR - READ CURRENT TAPE ADDRESS                                    *
*          WITH THE CURRENT ADDRESS AND THE STARTING RECORD NUMBER OF   *
*          THE FILE, WE CAN COMPUTE THE LOCATION OF THE FILE POINTER    *
*          IN ORDER TO CALCULATE HOW MANY REGISTERS WE HAVE LEFT FROM   *
*          THE FILE POINTER TO THE END OF THE FILE.                     *
*       1. ADDRESS CASSETTE AS A TALKER                                 *
*       2. SEND SEC.CMD - "READ ADDRESS"                                *
*       3. READ 2 BYTES OF RECORD # AND 1 BYTE OF BYTE #                *
*       4. COMPUTE NUMBER OF REGISTERS LEFT IN THE FILE                 *
* INPUT:   N[3:0] = CURRENT FILE SIZE IN NUMBER OF REGISTERS            *
* OUTPUT:  N[3:0] = NUMBER OF REGISTERS LEFT IN THE FILE                *
* USES:    A, B, C, +1 SUBROUTINE LEVEL                                 *
*************************************************************************
*
   646                   ENTRY  REQADR
*
   648 1100 REQADR     1 GOSUB  TALKER        ADDRESS DEVICE AS A TALKER
   648 1101            0                      *ILCAS&CTL:  CS0, @0262
   649 1102          460 LDI                  LOAD LOW 12 BITS OF C WITH
   650 1103          303 CON    @303          DDT 3 = READ ADDRESS
   651 1104            1 GOSUB  SCMD          SEND CMD - "READ ADDRESS"
   651 1105            0                      *ILCAS&CTL:  CS0, @0272
   652 1106          234 PT=    5             SET UP BYTE COUNTER
```

```
   653 1107           1 GOSUB  NATNRD        SEND READY COMMAND "NAT"
   653 1110           0                      *ILCAS&CTL:  CS0, @0352
   654 1111 REQA10    1 GOSUB  RDDFRM        READ 1ST BYTE OF RECORD #
   654 1112           0                      *ILCAS&CTL:  CS0, @0420
   655 1113        1756 A SL                 SHIFT REG A LEFT ONE BYTE
   656 1114        1756 A SL                  TO MAKE ROOM FOR NEW BYTE
   657 1115         266 AC EX  XS            C[XS]=LOW NYBBLE OF A BYTE
   658 1116         406 A=C    X             A[X]=LN OF BYTE + NEW BYTE
   659 1117        1724 DEC PT               DECREMENT BYTE COUNTER
   660 1120        1424 ? PT=  1             READ THE 4TH BYTE YET ?
   661 1121          37 GOC    REQA15 (1124) YES, THE 4TH BYTE IS ETO
   662 1122        1200 HPIL=C 2             NO, ECHO BYTE JUST READ
   663 1123        1663 GOTO   REQA10 (1111) READ NEXT FRAME
   664 1124 REQA15 1104 S9=    0             CLEAR THE ERROR FLAG
   665 1125         460 LDI                  LOAD LOW 12 BITS OF C WITH
   666 1126         100 CON    @100          @100 = END TRANSMISSION OK
   667 1127        1552 ? A#C  WPT           WAS LAST BYTE AN ETO ?
   668 1130        1467 GOC    RDKYER (1076) NO, GENERATE AN ERROR
   669 1131        1616 A SR                 YES, SHIFT REG A RIGHT ONE
   670 1132        1616 A SR                  BYTE TO REMOVE ETO BYTE
   671 1133         212 B=A    WPT           B[1:0] = BYTE NUMBER
   672 1134        1616 A SR                 SHIFT REG A RIGHT ONE BYTE
   673 1135        1616 A SR                  TO REMOVE THE BYTE NUMBER
   674 1136          32 A=0    M             A[2:0] = 3 DIGITS OF REC #
   675 1137        1740 RTN                  END OF REQADR ROUTINE
*
*****************************************************************
* DATSUB - FUNCTIONS LIKE WRTRX AND READRX USE THE X-REGISTER TO *
*          INDICATE STARTING AND ENDING REGISTERS.  THIS ROUTINE *
*          CHECKS THE STARTING AND ENDING REGISTERS, AND COMPUTES *
*          THE NUMBER OF REGISTERS NECESSARY TO READ OR WRITE.   *
*          THERE ARE TWO FATAL ERRORS THAT MIGHT HAPPEN:         *
*       1. STARTING OR ENDING REGISTER DOES NOT EXIST - "NONEXISTENT" *
*       2. FILE NOT LARGE ENOUGH TO HOLD THE REGISTERS - "NO ROOM"    *
*                                                               *
* INPUT  - X-REG   = BBB.EEE, WHERE BBB IS STARTING REGISTER NUMBER *
*                               AND EEE IS ENDING REGISTER NUMBER  *
*          N[11:8] = FILE SIZE IN NUMBER OF REGISTERS           *
* OUTPUT - M[X]    = NUMBER OF REGISTERS TO READ OR WRITE       *
*          M[M]    = STARTING REGISTER ADDRESS - 1              *
*                                                               *
* USES:    A, B[X], C, M, N, +1 SUBROUTINE LEVEL                *
*                                                               *
* DATALL - SAME AS DATSUB EXCEPT WILL WRITE ALL THE REGISTERS   *
*****************************************************************
*
   697                 ENTRY  DATSUB
   698                 ENTRY  DATALL
   699                 ENTRY  DATSER
*
*****************************************************************
* DFBDCK - CHECKS IF THE FOLLOWING READ OR WRITE WILL CROSS THE DATA *
*          FILE BOUNDARY                                        *
* INPUT:   REG.9[5:0] = CURRENT CASSETTE ADDRESS                *
*          N          = FILE ENTRY                              *
* OUTPUT:  M[X]       = NUMBER OF REGISTERS TO READ OR WRITE    *
*          M[5:3]     = STARTING REGISTER ADDRESS               *
*          REG.9[4:0] = NUMBER OF BYTES OF CURRENT ADDRESS PAST *
*                       BEGINNING OF FILE                       *
* USES:    A, B, C, N, +1 SUBROUTINE LEVEL                      *
```

```
*****************************************************************
   712                  ENTRY  DFBDCK
*
   714 1140 DFBDCK 1170 C=REGN 9             GET CURRENT ADDRESS
   715 1141        1074 RCR    2             ROTATE IT INTO POSITION
   716 1142         406 A=C    X             A[X] = CURRENT RECORD #
   717 1143        1434 PT=    1             POINT AT LOWEST BYTE
   718 1144         230 C=G                  SEE IF LAST OP WAS RD/WRT
   719 1145        1342 ? C#0  PT            WAS LAST OPER A WRITE ?
   720 1146          27 GOC    DFCK10 (1150) YES, DON'T DECREMENT REC #
   721 1147         646 A=A-1  X             NO, DECREMENT RECORD #
   722 1150 DFCK10  260 C=N                  C[11:8] = STARTING REC #
   723 1151         474 RCR    8             C[X] = STARTING REC #
   724 1152         706 A=A-C  X             A[X] = # OF RECS PAST BOF
   725 1153          37 GOC    DFCK20 (1156) NOT POINTING AT DATA FILE
   726 1154         374 RCR    10            C[X] = NUMBER OF RECORDS
   727 1155        1406 ? A<C  X             CROSSED FILE BOUNDARY ?
   728 1156 DFCK20    1 GOLNC  FLTYER        YES, DISPLAY: FL TYPE ERR
   728 1157           2                      *ILCAS&CTL:  CS2, @0212
   729 1160        1756 A SL                 NO, SHIFT LEFT ONE BYTE TO
   730 1161        1756 A SL                  MAKE ROOM FOR BYTE PTR
   731 1162        1170 C=REGN 9             C[1:0] = BYTE POINTER
   732 1163         412 A=C    WPT           A[4:0] = # BYTES PAST BOF
   733 1164         256 AC EX                C[4:0] = # BYTES PAST BOF
   734 1165        1150 REGN=C 9             SAVE IN REGISTER 9
*
   736 1166 DATSUB    1 GOSUB  FNDEND        FIND 1ST NON-EXIST REG ADR
   736 1167           0                      *MAINFRAME:  CN5, @1460
   737 1170         246 AC EX  X             COPY RESULT TO C[X]
   738 1171         530 M=C                  AND SAVE IT IN M[X]
   739 1172         106 C=0    X             CLEAR C[X]
   740 1173        1160 DADD=C               ENABLE CHIP 0
   741 1174         370 C=REGN 3             LOAD X-REGISTER
   742 1175           1 GOSUB  BCDBIN        CONVERT INT(X) TO BINARY
   742 1176           0                      *MAINFRAME:  CN0, @1343
   743 1177         246 AC EX  X             A[X]=REG # OF STARTING REG
   744 1200        1570 C=REGN 13            GET REGISTER 00 ADDRESS
   745 1201          74 RCR    3             C[X]=REGISTER 00 ADDRESS
   746 1202         506 A=A+C  X             A[X]=ABS ADDR OF START REG
   747 1203         630 C=M                  GET ADDRESS OF REG LIMIT
   748 1204        1406 ? A<C  X             1ST REGISTER OVER LIMIT ?
   749 1205          57 GOC    DATS05 (1212) NO, CONTINUE PROCESSING
   750 1206 DATSER    1 GOSUB  UNL           YES, SEND UNLISTEN COMMAND
   750 1207           0                      *ILCAS&CTL:  CS0, @0257
   751 1210           1 GOLONG ERRNE         ERROR MESSAGE: NONEXISTENT
   751 1211           2                      *MAINFRAME:  CN0, @1340
   752 1212 DATS05  206 B=A    X             B[X]=ABS ADDR OF START REG
   753 1213         370 C=REGN 3             READ X-REGISTER INTO REG C
   754 1214         416 A=C                  MAKE A COPY IN A-REGISTER
   755 1215        1240 SETDEC               REGISTER CONTENTS DECIMAL
   756 1216        1526 ? A#0  XS            IS X < 1 (EXP SIGN = 9) ?
   757 1217          47 GOC    DATS20 (1223) YES, DON'T DO SHIFTING
   758 1220 DATS10 1772 A SL   M             SHIFT OUT INT. PART OF X
   759 1221         646 A=A-1  X             REDUCE EXPONENT BY 1
   760 1222        1763 GONC   DATS10 (1220) KEEP GOING UNTIL EXP = 0
   761 1223 DATS20  460 LDI                  LOAD LOW 12 BITS OF C WITH
   762 1224           3 CON    3             3 TO BE ADDED TO EXPONENT
   763 1225         506 A=A+C  X             THUS MULTIPLYING BY 1000
   764 1226         256 AC EX                C = 1000 * FRAC(X)
   765 1227        1140 SETHEX               RETURN TO HEXADECIMAL MODE
```

```
   766 1230           1 GOSUB  BCDBIN         CONVERT TO BINARY
   766 1231           0                       *MAINFRAME:  CN0, @1343
   767 1232         406 A=C    X              A[X]=REL. ADDR OF END REG
   768 1233        1570 C=REGN 13             GET REGISTER 00 ADDRESS
   769 1234          74 RCR    3              C[X]=REGISTER 00 ADDRESS
   770 1235         506 A=A+C  X              A[X]=ABS ADDR OF END REG
   771 1236         630 C=M                   GET 1ST NON-EXIST REG ADDR
   772 1237        1406 ? A<C  X              END REGISTER OVER LIMIT ?
   773 1240        1463 GONC   DATSER (1206)  YES, SEND ERROR MESSAGE
   774 1241         606 A=A-B  X              B[X]=ABS ADDR OF START REG
   775 1242          37 GOC    DATS40 (1245)  NUMBER OF REGISTERS < 0
   776 1243        1506 ? A#0  X              NUMBER OF REGISTERS = 0 ?
   777 1244          27 GOC    DATS45 (1246)  NO, INCREMENT # OF REGS
   778 1245 DATS40    6 A=0    X              YES, SET # REGISTERS TO 0
   779 1246 DATS45  546 A=A+1  X              INCREMENT # OF REGISTERS
   780 1247        1610 S0=    1              CHECK EOF (END OF FILE)
   781 1250 DATS50  256 AC EX                 C[X]=# OF REGS TO READ/WRT
   782 1251          74 RCR    3              SAVE IN M[X]= # OF REGS
   783 1252         306 C=B    X                      M[M]= 1ST REG ADDR
   784 1253         674 RCR    11             ROTATE TO NORMAL POSITION
   785 1254         530 M=C                   SAVE TO M AS SHOWN ABOVE
   786 1255         132 C=0    M              C[M]=0, C[X]=# OF REGS
   787 1256           1 GOSUB  RG-BY#         # OF REGS TO # OF BYTES
   787 1257           0                       *ILCAS&CTL:  CS2, @1761
   788 1260        1170 C=REGN 9              GET # OF BYTES PAST BOF
   789 1261         516 A=A+C                 A[4:0]=TOTAL # BYTES F/BOF
   790 1262         212 B=A    WPT            B[4:0]=TOTAL # BYTES F/BOF
   791 1263         260 C=N                   C[3:0]=# OF REGS IN FILE
   792 1264           1 GOSUB  RG-BY#         # OF REGS TO # OF BYTES
   792 1265           0                       *ILCAS&CTL:  CS2, @1761
   793 1266        1452 ? A<B  WPT            WILL CROSS A BOUNDARY ?
   794 1267        1640 RTN NC                NO, RETURN TO CALLER
   795 1270        1614 ?S0=1                 NEED TO CHECK EOF ?
   796 1271           1 GOLC   CSEOF          YES, DISPLAY "END OF FILE"
   796 1272           3                       *ILCAS&CTL:  CS1, @1474
   797 1273         260 C=N                   READ WHAT IS IN THE FILE
   798 1274         406 A=C    X              A[X]=FILE SIZE IN REGS
   799 1275         630 C=M                   C[X]=OLD # OF REGS
   800 1276         246 AC EX  X              C[X]=NEW # OF REGS IN FILE
   801 1277         530 M=C                   SAVE 1ST REG, # REGS TO M
   802 1300        1740 RTN                   END OF DFBDCK/DATSUB
*
   804 1301 DATALL    1 GOSUB  FNDEND         FIND ADDRESS OF LAST REG
   804 1302           0                       *MAINFRAME:  CN5, @1460
   805 1303         116 C=0                   CLEAR ACCUMULATOR
   806 1304        1160 DADD=C                ENABLE CHIP 0
   807 1305        1150 REGN=C 9              WRITE STARTING FROM BOF
   808 1306        1570 C=REGN 13             C[5:3] = REG 00 ADDRESS
   809 1307          74 RCR    3              C[X] = REG 00 ADDRESS
   810 1310         346 B=C    X              B[X] = REG 00 ADDRESS
   810 1311         306                       (INSERTED BY ASSEMBLER)
   811 1312         706 A=A-C  X              A[X] = # OF REG TO WRITE
   812 1313        1506 ? A#0  X              IS SIZE EQUAL TO 0 ?
   813 1314        1347 GOC    DATS50 (1250)  NO, HANDLE THE REGISTERS
   814 1315           1 GOLONG DATSER         YES, UNLISTEN & ERROR
   814 1316           2                       *ILCAS&CTL:  CS3, @1206
*
*******************************************************************
* SVBREN - SAVE CURRENT BYTE POINTER AND READ LAST FILE ENTRY     *
*          FROM DIRECTORY BUFFER                                  *
```

```
* INPUT:   LAST ACCESSED FILE ENTRY POINTER IS SAVED IN DIRECTORY      *
*          BUFFER AT BYTE #250. IF THIS NUMBER = 250, LAST OPERATION   *
*          WAS NOT A READ OR WRITE TO A DATA FILE                      *
* OUTPUT:  REG.9[4:2] = CURRENT RECORD NUMBER                          *
*          REG.9[1:0] = CURRENT BYTE NUMBER                            *
*          G = BYTE #251 IN DIRECTORY BUFFER                           *
*              IF G=0, LAST OPERATION ON A DATA FILE WAS READ          *
*              IF G=1, LAST OPERATION ON A DATA FILE WAS WRITE         *
************************************************************************
   828                    ENTRY  SVBREN
*
   830 1317 SVBREN    1 GOSUB  REQADR         READ CURRENT ADDRESS
   830 1320           0                       *ILCAS&CTL:  CS3, @1100
   831 1321         256 AC EX                 C[3:0] = CURRENT RECORD #
   832 1322        1574 RCR    12             ROTATE REG C LEFT 1 BYTE
   833 1323         312 C=B    WPT            C[1:0] = CURRENT BYTE #
   834 1324        1150 REGN=C 9              WRITE OUTPUT TO REG.9[4:0]
   835 1325         460 LDI                   LOAD LOW 12 BITS OF C WITH
   836 1326         372 CON    250            BYTE POINTER VALUE OF 250
   837 1327           1 GOSUB  SETBPC         SET BYTE POINTER TO 250
   837 1330           0                       *ILCAS&CTL:  CS3, @1645
   838 1331           1 GOSUB  RDLPBK         SEND CMD - READ LOOP BACK
   838 1332           0                       *ILCAS&CTL:  CS1, @1652
   839 1333           1 GOSUB  NATNRD         SEND READY COMMAND "NAT"
   839 1334           0                       *ILCAS&CTL:  CS0, @0352
   840 1335           1 GOSUB  RDDFRM         READ LAST FILE ENTRY PTR
   840 1336           0                       *ILCAS&CTL:  CS0, @0420
   841 1337        1200 HPIL=C 2              ECHO THE BYTE JUST READ
   842 1340         346 BC EX  X              SAVE ENTRY POINTER IN B[X]
   843 1341           1 GOSUB  NRD            READ NEXT BYTE IN BUFFER
   843 1342           0                       *ILCAS&CTL:  CS0, @0364
   844 1343        1634 PT=    0              POINT AT LOWEST DIGIT
   845 1344         130 G=C                   SAVE THIS BYTE IN G
   846 1345         146 AB EX  X              GET LAST FILE ENTRY PTR
   847 1346         460 LDI                   LOAD LOW 12 BITS OF C WITH
   848 1347         372 CON    250            DATA VALUE OF 250
   849 1350        1546 ? A#C  X              ENTRY POINTER = 250 ?
   850 1351           1 GOLNC  FLTYER         YES, NOT A VALID ENTRY PTR
   850 1352           2                       *ILCAS&CTL:  CS2, @0212
   851 1353           1 GOSUB  SETBPL         PUT THE ENTRY POINTER BACK
   851 1354           0                       *ILCAS&CTL:  CS3, @1627
   852 1355           1 GOSUB  RDLPBK         GO TO READ LOOP BACK MODE
   852 1356           0                       *ILCAS&CTL:  CS1, @1652
   853 1357           1 GOSUB  RENT10         READ NEXT FILE ENTRY
   853 1360           0                       *ILCAS&CTL:  CS2, @0243
   854 1361           1 GOLONG RSTBP          RESTORE BYTE POINTER
   854 1362           2                       *ILCAS&CTL:  CS3, @1644
************************************************************************
* SVENTR - SAVE FILE ENTRY PTR IN DIR BUFFER BYTE #250 AND REMEMBER    *
*          LAST OPERATION TO A DATA FILE WAS A READ IN BYTE #251.      *
* SVENTW - SAME AS SVENTR EXCEPT LAST OPERATION IS WRITE.              *
************************************************************************
   860                    ENTRY  SVENTR
   861                    ENTRY  SVENTW
   862                    ENTRY  SVMODE
   863 1363 SVENTR 1604 S0=    0              S0=0: LAST OPER WAS READ
   864 1364           23 GOTO   SVENT  (1366) GET CURRENT TAPE ADDRESS
   865 1365 SVENTW 1610 S0=    1              S0=1: LAST OPER WAS WRITE
   866 1366 SVENT     1 GOSUB  REQADR         GET CURRENT TAPE ADDRESS
   866 1367           0                       *ILCAS&CTL:  CS3, @1100
```

```
   867 1370         1410 S1=    1             NOT ONLY SAVE MODE
   868 1371          146 AB EX  X             A[X] = BYTE NUMBER
   869 1372          460 LDI                  LOAD LOW 12 BITS OF C WITH
   870 1373           40 CON    32            NUMBER OF BYTES TO BACK UP
   871 1374          706 A=A-C  X             BACK UP 32 BYTES
   872 1375          206 B=A    X             B[X] = BYTE NUMBER
   873 1376          460 LDI                  LOAD LOW 12 BITS OF C WITH
   874 1377          372 CON    250           BYTE POINTER VALUE OF 250
   875 1400 SVEN10     1 GOSUB  SETBPC        SET BYTE POINTER TO 250
   875 1401            0                      *ILCAS&CTL:  CS3, @1645
   876 1402            1 GOSUB  WRLPBK        SEND CMD - WRITE LOOP BACK
   876 1403            0                      *ILCAS&CTL:  CS1, @1660
   877 1404         1414 ?S1=1                ONLY SAVE MODE ?
   878 1405           43 GONC   SVEN20 (1411) YES, DON'T SEND BYTE PTR
   879 1406          306 C=B    X             C[X]=CURR ENTRY BYTE PTR
   880 1407            1 GOSUB  SDATA0        SEND CURR ENTRY BYTE PTR
   880 1410            0                      *ILCAS&CTL:  CS0, @0446
   881 1411 SVEN20   106 C=0    X             CLEAR C[X]
   882 1412         1614 ?S0=1                LAST OPER WAS A WRITE ?
   883 1413           23 GONC   *+2    (1415) NO, IT WAS A READ
   884 1414         1046 C=C+1  X             INCREMENT C[X]
   885                   LEGAL                (CLEAR THE CARRY FLAG)
   886 1415            1 GOLONG SDATA0        SEND BYTE, EITHER 0 OR 1
   886 1416            2                      *ILCAS&CTL:  CS0, @0446
*
   888 1417 SVMODE  1404 S1=    0             S1=0: SAVE MODE
   889 1420          460 LDI                  LOAD LOW 12 BITS OF C WITH
   890 1421          373 CON    251           BYTE POINTER = 251
   891 1422         1563 GOTO   SVEN10 (1400) SET BYTE PTR, WRT LOOP BK
*
*****************************************************************
* INTCAL - CALCULATE INTEGER = INT [A*C + 0.5]                  *
*                                                               *
* USES:    A, B, C, M, PT, S5, 2 SUBROUTINE LEVELS              *
* INPUT:   A= (NNN-1)/(MAX-MIN) = FLOATING POINT [INTCAL ONLY]  *
*          C= (X-MIN) OR (-MIN) [INTCAL]                        *
*          DECIMAL MODE [INTCAL ONLY]                           *
* OUTPUT:  C[X] = BINARY NUMBER                                 *
*          HEX MODE, DOESN'T CHANGE CHIP ENABLE                 *
*****************************************************************
   903                   ENTRY  INTCAL
   904 1423 INTCAL     1 GOSUB  MP2-10        (  )(NNN-1)/(MAX-MIN)
   904 1424            0                      *MAINFRAME:  CN6, @0115
* CAN'T OVERFLOW, AND UNDERFLOW RTNS C=0 WHICH IS OK, SO DON'T CHECK.
   906
   907 1425            1 GOSUB  OVFL10        NORMAL UNDERFLOW TO 0
   907 1426            0                      *MAINFRAME:  CN5, @0051
   908 1427          416 A=C                  A=(  )(NNN-1)/(MAX-MIN)
   909 1430           36 A=0    S             TAKE ABSOLUTE VALUE
   910 1431          116 C=0                  CLEAR ACCUMULATOR
   911 1432         1246 C=-C-1 X             EXPONENT = -1
   912 1433          520 LC     5             C = 0.5
   913 1434            1 GOSUB  AD2-10        C = A + C
   913 1435            0                      *MAINFRAME:  CN6, @0007
* CAN'T OVERFLOW SINCE 0.5 ADDS NOTHING TO "9 E99"
   915
   916 1436          210 S5=    1             GET INTEGER PART
   917 1437            1 GOSUB  INTFRC        INT [C + 0.5]
   917 1440            0                      *MAINFRAME:  CN6, @0473
   918 1441         1140 SETHEX               GO TO HEXADECIMAL MODE
```

```
   919                    ENTRY  CONV3D
   920                    ENTRY  CONV3C
   921 1442 CONV3C  406 A=C    X             COPY EXPONENT TO A[X]
   922 1443         136 C=0    S             TAKE ABSOLUTE VALUE
   923 1444         106 C=0    X             INITIALIZE ANSWER TO 0
   924 1445        1372 ? C#0  M             HANDLES ZERO UNNORMAL #S
   925 1446        1640 RTN NC               RETURN IF ZERO MANTISSA
   926 1447        1526 ? A#0  XS            NEGATIVE EXPONENT ?
   927 1450        1540 RTN C                YES, ANSWER IS 000
   928 1451        1574 RCR    12            MOVE 1ST DIGIT TO C[0]
   929 1452         646 A=A-1  X             WAS EXPONENT 0 ?
   930 1453         107 GOC    XGOTI  (1463) YES, CONVERT TO BINARY
   931 1454        1374 RCR    13            ROTATE NEXT DIGIT IN
   932 1455         646 A=A-1  X             WAS EXPONENT 1 ?
   933 1456          57 GOC    XGOTI  (1463) YES, CONVERT TO BINARY
   934 1457        1374 RCR    13            MOVE THIRD DIGIT IN
   935 1460         646 A=A-1  X             WAS EXPONENT 2 ?
   936 1461           1 GOLNC  ERRDE         NO, NUMBER TOO LARGE
   936 1462           2                      *MAINFRAME: CN10, @0055
   937 1463 XGOTI     1 GOLONG GOTINT        CONVERT ANSWER TO BINARY
   937 1464           2                      *MAINFRAME:  CN0, @1370
*
*************************************************************************
* CONV3D - CONVERTS THE THREE DIGITS TO THE LEFT OF THE DECIMAL POINT   *
*          IN THE X-REGISTER TO A BINARY NUMBER AND LEAVES IT IN C[X].  *
* CONV3C - SAME EXCEPT INPUT IS IN C.                                   *
*          IGNORES SIGN OF X. IF ABS(X) IS > 999, GIVES "DATA ERROR".   *
*          IF X CONTAINS AN ALPHA STRING, GIVES "ALPHA DATA" ERROR.     *
*          IF NON-NORMALIZED NUMBER WITH ZERO MANTISSA, RETURNS ZERO.   *
*                                                                       *
* ASSUMES: CHIP 0 ENABLED AND HEX MODE.                                 *
* USES:    A[X] AND C, NO PT, NO STS, 1 ADDITIONAL SUB LEVEL (GOTINT).  *
*          USUALLY EXITS VIA GOTINT IN THE MAINFRAME.                   *
*          RETURNS ANSWER IN C[X].                                      *
*************************************************************************
   952 1465 CONV3D    1 GOSUB  ACKX          GET X, ERROR IF ALPHA
   952 1466           0                      *ILCAS&CTL:  CS0, @1760
   953 1467        1533 GOTO   CONV3C (1442) X IN C, READY FOR CONV3C
*
*
*
*************************************************************************
* PCHKKB - CHECK KEYBOARD (PRINTER FUNCTION)                            *
*                                                                       *
* RETURNS IF NEITHER THE "R/S" KEY NOR THE "ON" KEY IS DOWN             *
*                                                                       *
* INPUT:   NONE                                                         *
* OUTPUT:  NONE                                                         *
* USES:    A[X], C, NO PT, NO STS, NO ADDITIONAL SUBROUTINE LEVELS      *
*************************************************************************
   966                    ENTRY  PCHKKB
   967                    ENTRY  PCKBRT
   968 1470 PCHKKB 1714 CHK KB               IS A KEY PRESSED DOWN ?
   969 1471        1640 RTN NC               NO KEY PRESSED, RETURN
   970 1472        1040 C=KEYS               YES, GET KEY CODE
   971 1473          74 RCR    3             KEY CODE TO C[X]
   972 1474         126 C=0    XS            CLEAR C[2] (EXPONENT SIGN)
   973 1475         406 A=C    X             KEY CODE TO A[X]
   974 1476         460 LDI                  LOAD LOW 12 BITS OF C WITH
   975 1477          30 CON    24            HEX 18 = "OFF" KEY
```

```
   976 1500         1546 ? A#C  X              "OFF" KEY HIT ?
   977 1501           57 GOC    PCKB10 (1506) NO, ANOTHER KEY PRESSED
   978 1502            1 GOSUB  IFC            SEND INTERFACE CLEAR CMD
   978 1503            0                       *ILCAS&CTL:  CS3, @0113
   979 1504            1 GOLONG OFF            TURN CALCULATOR OFF
   979 1505            2                       *MAINFRAME:  CN4, @0710
   980 1506 PCKB10   460 LDI                   LOAD LOW 12 BITS OF C WITH
   981 1507          207 CON    135            HEX 87 = "R/S" KEY
   982 1510         1546 ? A#C  X              "R/S" KEY HIT ?
   983 1511           67 GOC    OUTPCK (1517) NO, ANOTHER KEY PRESSED
   984 1512 PCKBRT 1304 S13=   0              YES, CLEAR RUNNING FLAG
   985 1513            1 GOSUB  IFC            SEND INTERFACE CLEAR CMD
   985 1514            0                       *ILCAS&CTL:  CS3, @0113
   986 1515            1 GOLONG NFRKB          RESET THE KEYBOARD
   986 1516            2                       *MAINFRAME:  CN0, @0307
   987 1517 OUTPCK 1710 RST KB                 NO, TRY TO CLEAR THE KEY
   988 1520         1740 RTN                   END OF CHECK KEYBOARD
*
*****************************************************************
* CKANGL - CHECK IF CHARACTER WAS AN ANGLE SIGN (ASCII 0D)       *
*          IF IT IS, REPLACE 0D BY 7C (VERTICAL BAR CHARACTER: "|")    *
*          THIS IS DONE BECAUSE ASCII 0D IS AN ANGLE SIGN IN HELIOS    *
*          BUT PIL PRINTER WILL USE 0D AS CARRIAGE RETURN, SO WE       *
*          MUST MOVE THE ANGLE SIGN FROM 13(0D) TO 124(7C).            *
*                                                                      *
* USES:    B[X], N, PT, +0 SUBROUTINE LEVELS                           *
*****************************************************************
   999                   ENTRY  CKANGL
  1000                   ENTRY  CKANGB
  1001                   ENTRY  CKANGN
*
  1003 1521 CKANGL  206 B=A    X              SAVE A[X] IN B[X]
  1004 1522 CKANGB  160 N=C                   SAVE C INTO N
  1005 1523 CKANGN  406 A=C    X              PUT C[X] INTO A[X]
  1006 1524         1434 PT=    1             POINT AT LOWEST BYTE
  1007 1525          460 LDI                  LOAD LOW 12 BITS OF C WITH
  1008 1526           15 CON    13            13 (0D): HELIOS ANGLE SIGN
  1009 1527         1552 ? A#C  WPT            IS IT AN ANGLE SIGN ?
  1010 1530          213 GONC   CKANG5 (1551) YES, CONVERT TO 124 (7C)
  1011 1531          644 C=HPIL 6             READ DEVICE REGISTER 6
  1011 1532          672                      (INSERTED BY ASSEMBLER)
  1011 1533          603                      (INSERTED BY ASSEMBLER)
  1012 1534         1166 C=C-1  XS             PUT F INTO C[XS]
  1013 1535         1046 C=C+1  X              ARE WE TALKING TO T.V. ?
  1014 1536          103 GONC   CKANG3 (1546) NO, RESTORE REGS & RETURN
  1015 1537          460 LDI                  LOAD LOW 12 BITS OF C WITH
  1016 1540          176 CON    126           126(7E): HELIOS SIGMA SIGN
  1017 1541         1552 ? A#C  WPT            IS THIS THE SIGMA SIGN ?
  1018 1542           47 GOC    CKANG3 (1546) NO, RESTORE REGS & RETURN
  1019 1543          460 LDI                  LOAD LOW 12 BITS OF C WITH
  1020 1544           34 CON    28            REPLACEMENT ASCII CODE 28
  1021 1545           63 GOTO   CKANG6 (1553) RESTORE REGISTERS & RETURN
  1022 1546 CKANG3  260 C=N                   RESTORE C FROM N
  1023 1547 CKANG4  146 AB EX  X              RESTORE A[X] FROM B[X]
  1024 1550         1740 RTN                  RETURN TO CALLING ROUTINE
  1025 1551 CKANG5  460 LDI                   LOAD LOW 12 BITS OF C WITH
  1026 1552          174 CON    124           REPLACE 13(0D) BY 124(7C)
  1027 1553 CKANG6  406 A=C    X              SAVE C[X] IN A[X]
  1028 1554          260 C=N                  RESTORE C FROM N
  1029 1555          252 AC EX  WPT            LOWEST BYTE OF A TO C[1:0]
```

```
  1030 1556        1713 GOTO   CKANG4 (1547) RESTORE A[X] AND RETURN
*************************************************************************
* SEEK  -  SEEK TO A GIVEN RECORD                                       *
* INPUT:   A[3:0] = RECORD NUMBER                                       *
* OUTPUT:  CASSETTE AS A TALKER                                         *
* SEEKN  - SAME AS SEEK EXCEPT INPUT IS IN N[3:0]                       *
* SEEKRD - SEEK TO A GIVEN RECORD AND READ IT                           *
* INPUT:   SAME AS SEEK                                                 *
* OUTPUT:  CASSETTE AS A TALKER                                         *
*                                                                       *
* USES:    A, C, S[7:0], +1 SUBROUTINE LEVEL                            *
*************************************************************************
  1042                   ENTRY  SEEK
  1043                   ENTRY  SEEKC
  1044                   ENTRY  SEEKN
  1045                   ENTRY  SEEKRC
  1046                   ENTRY  SEEKRD
  1047                   ENTRY  SEEKRN
  1048                   ENTRY  SEEK40
  1049                   ENTRY  SETBPT
  1050                   ENTRY  SETBPL
  1051                   ENTRY  SEEKR2
  1052                   ENTRY  SETBP0
*
  1054 1557 SEEKR2  460 LDI                  LOAD LOW 12 BITS OF C WITH
  1055 1560           2 CON    2              RECORD NUMBER 2
  1056 1561         103 GOTO   SEEKRC (1571) SEEK TO RECORD NUMBER IN C
  1057 1562 SEEKN   260 C=N                  GET REC NUMBER FROM REG N
  1058 1563         474 RCR    8              ROTATE REC # INTO POSITION
  1059 1564 SEEKC   416 A=C                  COPY REC # FROM C INTO A
  1060 1565 SEEK    404 S8=    0              S8=0: SEEK ONLY, NO READ
  1061 1566          53 GOTO   SEEK10 (1573) GOTO SEND SEEK COMMAND
  1062 1567 SEEKRN  260 C=N                  GET REC NUMBER FROM REG N
  1063 1570         474 RCR    8              ROTATE REC # INTO POSITION
  1064 1571 SEEKRC  416 A=C                  COPY REC # FROM C INTO A
  1065 1572 SEEKRD  410 S8=    1              S8=1: SEEK, THEN READ REC
  1066 1573 SEEK10    1 GOSUB  LISTEN         SEND SEC.CMD - "SEEK"
  1066 1574           0                      *ILCAS&CTL:  CS0, @0335
  1067 1575         460 LDI                  LOAD LOW 12 BITS OF C WITH
  1068 1576         244 CON    @244           @244 = DEV DEP LISTENER 4
  1069 1577           1 GOSUB  SCMD           SEND SEC.CMD DDL4 - "SEEK"
  1069 1600           0                      *ILCAS&CTL:  CS0, @0272
  1070 1601         256 AC EX                SEND RECORD # (TWO BYTES)
  1071 1602         132 C=0    M              MAKE SURE REC.# < 512
  1072 1603        1074 RCR    2              ROTATE C[3:2] TO C[1:0]
  1073 1604         416 A=C                  SAVE C-REGISTER TO REG A
  1074 1605           1 GOSUB  SDATA0         SEND 1ST BYTE OF RECORD #
  1074 1606           0                      *ILCAS&CTL:  CS0, @0446
  1075 1607         256 AC EX                RESTORE C-REGISTER
  1076 1610        1574 RCR    12             ROTATE 2ND BYTE TO C[1:0]
  1077 1611           1 GOSUB  SDATA          SEND 2ND BYTE OF RECORD #
  1077 1612           0                      *ILCAS&CTL:  CS0, @0450
  1078 1613 SEEK20    1 GOSUB  CSSTAS         READ CASSETTE STATUS
  1078 1614           0                      *ILCAS&CTL:  CS0, @0700
  1079 1615           1 GOSUB  PLERCK         CHECK FOR ANY ERRORS
  1079 1616           0                      *ILCAS&CTL:  CS1, @1747
  1080 1617         214 ?S5=1                STILL BUSY ?
  1081 1620        1737 GOC    SEEK20 (1613) YES, KEEP TRYING STATUS
  1082 1621         114 ?S4=1                ANY ERROR ?
  1083 1622           1 GOLC   CSERR          YES, DISP CASSETTE ERROR
```

```
  1083 1623           3                       *ILCAS&CTL:  CS1, @0042
  1084 1624 SEEK30   414 ?S8=1                 NEED TO READ ?
  1085 1625          137 GOC    SEEK40 (1640) YES, READ A RECORD
  1086 1626 SETBP0     6 A=0    X             SET BYTE POINTER TO 00
  1087 1627 SETBPL     1 GOSUB  LISTEN        ADDRESS DEVICE AS LISTENER
  1087 1630            0                       *ILCAS&CTL:  CS0, @0335
  1088 1631 SETBPT   460 LDI                   LOAD LOW 12 BITS OF C WITH
  1089 1632          243 CON    @243          @243 = DEV DEP LISTENER 3
  1090 1633            1 GOSUB  SCMD          SEND DDL3 - "SET BYTE PTR"
  1090 1634            0                       *ILCAS&CTL:  CS0, @0272
  1091 1635          246 AC EX  X             C[X] = BYTE POINTER
  1092 1636            1 GOLONG SDATA0        SEND BYTE POINTER
  1092 1637            2                       *ILCAS&CTL:  CS0, @0446
  1093 1640 SEEK40   460 LDI                   LOAD LOW 12 BITS OF C WITH
  1094 1641          302 CON    @302          @302 = DEV DEP TALKER 2
  1095 1642            1 GOSUB  SCMDWT        SEND DDT2 - "READ"
  1095 1643            0                       *ILCAS&CTL:  CS1, @0030
*
*****************************************************************************
* RSTBP  - RESTORE BYTE POINTER                                             *
* INPUT:   REG.9[2:0] = BYTE POINTER                                        *
* ASSUMES: CHIP 0 ENABLE                                                    *
* USES:    A, X, C, +1 SUBROUTINE LEVEL                                     *
*****************************************************************************
  1103                     ENTRY  RSTBP
  1104                     ENTRY  SETBPC
*
  1106 1644 RSTBP   1170 C=REGN 9             GET BYTE POINTER FROM R9
  1107 1645 SETBPC   406 A=C    X             SAVE C[X] INTO A[X]
  1108 1646         1613 GOTO   SETBPL (1627) SET BYTE POINTER W/LISTEN
*
*****************************************************************************
* CNTBYT - COMPUTE NUMBER OF BYTES BETWEEN TWO ADDRESSES IN MM FORM         *
* INPUT:   A[3:0] = STARTING ADDRESS DRRR = DIGIT# REGISTER# START          *
*          B[3:0] = ENDING ADDRESS   DRRR = DIGIT# REGISTER# END            *
*          PT = 3                                                           *
* OUTPUT:  A[2:0] = NUMBER OF BYTES BETWEEN THE TWO ADDRESSES               *
* USES:    A[3:0], B[X], C[X], +0 SUBROUTINE LEVELS                         *
*****************************************************************************
  1118                     ENTRY  CNTBYT
*
  1120 1647 CNTBYT   116 C=0                   CLEAR BYTE COUNTER
  1121 1650          602 A=A-B  PT             END BYTE# > START BYTE# ?
  1122 1651           43 GONC   CBYT10 (1655) NO, CALC BYTE DIFFERENCE
  1123 1652          646 A=A-1  X              YES, DECREMENT START REG#
  1124 1653          642 A=A-1  PT             DEC, ADJ HEX BY 2 DIGITS
  1125 1654 CBYT05   642 A=A-1  PT             DECREMENT STARTING DIGIT#
  1126 1655 CBYT10   642 A=A-1  PT             DEC, REACH DIGIT 0 YET ?
  1127 1656           37 GOC    CBYT20 (1661) YES, NOW COUNT REGISTERS
  1128 1657         1046 C=C+1  X              NO, INCREMENT BYTE COUNT
  1129                   LEGAL                 (CLEAR THE CARRY FLAG)
  1130 1660         1743 GOTO   CBYT05 (1654) KEEP COUNTING DIGITS
  1131 1661 CBYT20   606 A=A-B  X              # OF REGISTERS DIFFERENCE
  1132 1662          346 BC EX  X              B[X]=# OF EXTRA BYTES
  1133 1663          246 C=A    X              C[X]=# OF REGS DIFFERENCE
  1133 1664          406                       (INSERTED BY ASSEMBLER)
  1134 1665          746 C=C+C  X              DOUBLE C[X] ONE TIME
  1135 1666          746 C=C+C  X              DOUBLE C[X] TWO TIMES
  1136 1667          746 C=C+C  X              DOUBLE C[X] THREE TIMES
  1137 1670          246 AC EX  X              EXCHANGE A[X] AND C[X]
```

```
  1138 1671         706 A=A-C  X             SUB C[X] (= MUL BY 7)
  1139 1672         446 A=A+B  X             ADD NUMBER OF EXTRA BYTES
  1140 1673        1740 RTN                  END OF CNTBYT ROUTINE
*
*
*************************************************************************
* SNBYTS - ROUTINE TO SEND A GIVEN NUMBER OF BYTES IN C-REGISTER        *
* ASSUMES: CASSETTE IS IN THE MIDDLE OF RECEIVING DATA                  *
* INPUT:   PT      = NUMBER OF BYTES TO SEND - 1                        *
*          C[13:0] = LEFT-JUSTIFIED BYTES TO SEND                       *
* USES:    A, C, PT, +1 SUBROUTINE LEVEL                                *
*************************************************************************
  1150                   ENTRY  SNBYTS
*
  1152 1674 SNBYTS 1574 RCR    12           C[1:0]=PREV LEFTMOST BYTE
  1153 1675         416 A=C                 SAVE C-REGISTER TO REG A
  1154 1676           1 GOSUB  SDATA0        SEND DATA BYTE FROM C[1:0]
  1154 1677           0                      *ILCAS&CTL:  CS0, @0446
  1155 1700        1624 ? PT=  0             SENT LAST BYTE YET ?
  1156 1701        1540 RTN C                YES, RETURN TO CALLER
  1157 1702        1724 DEC PT               NO, DECREMENT POINTER
  1158 1703         256 AC EX                RESTORE C-REG FROM REG A
  1159 1704        1703 GOTO   SNBYTS (1674) PROCESS NEXT DATA BYTE
*
*
*************************************************************************
* PWRDN  - SEND COMMAND "GROUP POWER DOWN"                              *
*************************************************************************
*
  1166                   ENTRY  PWRDN
*
  1168 1705         216 CON    @216          N
  1169 1706           4 CON    @04           D
  1170 1707          22 CON    @22           R
  1171 1710          27 CON    @27           W
  1172 1711          20 CON    @20           P
  1173 1712 PWRDN     1 GOSUB  PILEN         TURN THE CLOCK ON
  1173 1713           0                      *ILCAS&CTL:  CS0, @0143
  1174 1714           1 GOSUB  PLERCK        CHECK FOR ANY ERRORS
  1174 1715           0                      *ILCAS&CTL:  CS1, @1747
  1175 1716         344 HPL=CH 3             WRITE PARALLEL POLL REG 3
  1176 1717           1 CH=    @000          SHUT OFF AUTO IDY
  1177 1720         460 LDI                  LOAD LOW 12 BITS OF C WITH
  1178 1721         233 CON    @233          HEX 9B = LPD = LOOP PWR DN
  1179 1722           1 GOLONG SCMD          SEND LPD CMD TO HPIL LOOP
  1179 1723           2                      *ILCAS&CTL:  CS0, @0272
*
*************************************************************************
* BINBCD - CONVERT FROM BINARY TO BCD                                   *
*                                                                       *
* CONVERTS THE BINARY NUMBER IN A[X] TO BCD                             *
*                                                                       *
* USES:    A, B[X], C, NO ST, ACTIVE PT, 1 ADDITIONAL SUBROUTINE LEVEL *
*                                      (GENNUM CALLS SUBROUTINE)        *
* INPUT:   [BINBCD] A[X] = BINARY NUMBER,  A[S] = # OUTPUT DIGITS       *
*          [BINBD]  C[X] = BINARY NUMBER,  C[S] = # OUTPUT DIGITS       *
*          BOTH ENTRIES: HEX MODE                                       *
*                                                                       *
* OUTPUT:  A[M] = DIGIT STRING (LEFT-JUSTIFIED), B[S] = OUTPUT DIGITS   *
*          HEX MODE, LCD DISABLED, RAM DISABLED                         *
```

```
*************************************************************************
  1195                    ENTRY  BINBD0
  1196                    ENTRY  BINBDC
  1197                    ENTRY  BINBCD
  1198 1724 BINBD0  136 C=0    S             OUTPUT 2, 3, OR 4 DIGITS
  1199 1725 BINBDC  416 A=C                  INPUTS TO "A"
  1200 1726 BINBCD  460 LDI                  LOAD LOW 12 BITS OF C WITH
  1201 1727          20 CON    20            ADDR= CHIP 1 (NONEXISTENT)
  1202 1730        1160 DADD=C               UNSELECT RAM
  1203 1731         106 C=0    X             CLEAR C[X]
  1204 1732        1760 PFAD=C               UNSELECT PERIPHERALS
  1205 1733           1 GOLONG GENNUM        CALC BCD # & # DIGITS OUT
  1205 1734           2                      *MAINFRAME:  CN1, @0750
*
*
  1208
  1209                    FILLTO @1734
  1210                    EJECT
```

```
*
*************************************************************************
* DSWKUP - LOGIC FOR NORMAL WAKE UP FROM DEEP SLEEP                     *
*************************************************************************
  1215 1735 DSWKUP  530 M=C                    SAVE C-REGISTER
  1216 1736           1 GOSUB  WKUPLP          WAKE UP THE LOOP
  1216 1737           0                        *ILCAS&CTL:  CS1, @1706
  1217 1740        1114 ?S9=1                  LOOP INTAKE ?
  1218 1741         167 GOC    KYCKX1 (1757)  NO, DON'T CHECK DEVICES
  1219 1742         444 C=HPIL 4              GET SELECTED LOOP ADDRESS
  1219 1743         472                       (INSERTED BY ASSEMBLER)
  1219 1744         403                       (INSERTED BY ASSEMBLER)
  1220 1745         404 S8=    0              DON'T DISPLAY ERROR MSG
  1221 1746           1 GOSUB  CHKLAD          VERIFY THE ADDRESS
  1221 1747           0                        *ILCAS&CTL:  CS0, @1310
  1222 1750           1 GOSUB  IFC             SEND INTERFACE CLEAR CMD
  1222 1751           0                        *ILCAS&CTL:  CS3, @0113
  1223 1752           1 GOSUB  FNDPTR          LOCATE THE PRINTER
  1223 1753           0                        *ILCAS&CTL:  CS0, @0575
  1224 1754          33 GOTO   KYCKX1 (1757)  PRINTER WAS NOT FOUND
  1225 1755           1 GOSUB  SF5521          SET FLAGS 55&21 & AUTOIDY
  1225 1756           0                        *ILCAS&CTL:  CS0, @1343
  1226 1757 KYCKX1 1670 C=REGN 14              C[1:0] = FLAGS 48-55
  1227 1760        1530 ST=C                   RESTORE STATUS SET 0
  1228 1761         630 C=M                    RESTORE C-REGISTER
  1229 1762           1 GOLONG RMCK10          ROM CHECK SUBROUTINE
  1229 1763           2                        *MAINFRAME:  CN9, @1763
  1230                   FILLTO @1763           FIX ROM TABLE LOCATION
  1231 1764 PPAUSE    0 NOP                    ENTRY FROM PAUSE LOOP
  1232 1765 PRUN      0 NOP                    MAIN RUNNING LOOP
  1233 1766 WAKEP  1473 GOTO   DSWKUP (1735)  WAKEUP F/DEEP SLEEP NO KEY
  1234 1767 POWOFP    0 NOP                    POWER OFF ENTRY POINT
  1235 1770 I/OSVP    0 NOP                    I/O SERVICE ENTRY POINT
  1236 1771 DEEPSP 1443 GOTO   DSWKUP (1735)  WAKEUP FROM DEEP SLEEP
  1237 1772 COLDSP 1433 GOTO   DSWKUP (1735)  COLD START ENTRY POINT
  1238 1773 PRTID    10 CON    @10            H
  1239 1774           61 CON    @61            1
  1240 1775           23 CON    @23            S
  1241 1776            3 CON    @03            C
  1242 1777 CKSUMP    0 NOP                    PRINTER CHECKSUM
*
  1244                   UNLIST
  1247                   END

  ERRORS :      0
```

SYMBOL TABLE (SCPL4B - ILCAS&CTL QUAD 3 = CS3 = ADDRESSES @76000-77777)

```
BINBCD  1726   -
BINBD0  1724   -
BINBDC  1725   -
CBYT05  1654   -  1660
CBYT10  1655   -  1651
CBYT20  1661   -  1656
CHKCS0   336   -
CHKCST   335   -
CHKPCT   414   -
CKANG3  1546   -  1542 1536
CKANG4  1547   -  1556
CKANG5  1551   -  1530
CKANG6  1553   -  1545
CKANGB  1522   -
CKANGL  1521   -
CKANGN  1523   -
CKSUMP  1777   -
CNTBYT  1647   -
COLDER   661   -
COLDSP  1772   -
CONV3C  1442   -  1467
CONV3D  1465   -
COPYBF   306   -   363
CSERE0   400   -
CSEREX   376   -   434  413
CSNOFD   364   -   340
CSRDY    341   -   347
CSRDY1   350   -   342
DATALL  1301   -
DATS05  1212   -  1205
DATS10  1220   -  1222
DATS20  1223   -  1217
DATS40  1245   -  1242
DATS45  1246   -  1244
DATS50  1250   -  1314
DATSER  1206   -  1240
DATSUB  1166   -
DEEPSP  1771   -
DFBDCK  1140   -
DFCK10  1150   -  1146
DFCK20  1156   -  1153
DSPERR   405   -
DSWKUP  1735   -  1772 1771 1766
FINDID     6   -
FLPT10   420   -
FNID10    13   -    52
FNID20    35   -    40
FNID30    41   -    36
FNID35    45   -    34
FNID40    57   -    51
FNID45    53   -    44
FNID60    62   -
FNID65    63   -    56
FNID70   101   -    76
FNIDRT    60   -   102
I/OSVP  1770   -
```

```
IFC      113   -
INTCAL  1423   -
INTDIR   361   -
KYCKX1  1757   -  1754 1741
OUTPCK  1517   -  1511
PCHKKB  1470   -
PCKB10  1506   -  1501
PCKBRT  1512   -
POWOFP  1767   -
PPAUSE  1764   -
PRTID   1773   -
PRUN    1765   -
PURGEF   135   -
PURGEP   140   -
PWRDN   1712   -
RALL10   665   -   660
RDKY10  1025   -  1037
RDKY15  1032   -  1062
RDKY20  1035   -  1043
RDKY30  1040   -  1033
RDKY40  1044   -  1031
RDKY50  1063   -  1045
RDKYER  1076   -  1130
READA    606   -
READK   1001   -
REQA10  1111   -  1123
REQA15  1124   -  1121
REQADR  1100   -
REWENT   145   -
RSTBP   1644   -
SEEK    1565   -
SEEK10  1573   -  1566
SEEK20  1613   -  1620
SEEK30  1624   -
SEEK40  1640   -  1625
SEEKC   1564   -
SEEKN   1562   -
SEEKR2  1557   -
SEEKRC  1571   -  1561
SEEKRD  1572   -
SEEKRN  1567   -
SETBP0  1626   -
SETBPC  1645   -
SETBPL  1627   -  1646
SETBPT  1631   -
SNBYTS  1674   -  1704
STOPIO   111   -
SVBREN  1317   -
SVEN10  1400   -  1422
SVEN20  1411   -  1405
SVENT   1366   -  1364
SVENTR  1363   -
SVENTW  1365   -
SVMODE  1417   -
VERF10   453   -   462
VERIFY   443   -
WAKEP   1766   -
WRET10   160   -   151
WRET15   171   -
WRET30   227   -   222
```

WRET40	231	-	226	220
WRET50	274	-	203	
WRTA	467	-		
WRTA10	500	-	506	
WRTA15	543	-	540	
WRTA20	571	-		
WRTK	704	-		
WRTK10	717	-	727	
WRTK15	731	-	723	
WRTK20	761	-	735	
XGOTI	1463	-	1456	1453

ENTRY TABLE (SCPL4B - ILCAS&CTL QUAD 3 = CS3 = ADDRESSES @76000-77777)

```
BINBCD  1726   -
BINBD0  1724   -
BINBDC  1725   -
CHKCS0   336   -
CHKCST   335   -
CHKPCT   414   -
CKANGB  1522   -
CKANGL  1521   -
CKANGN  1523   -
CNTBYT  1647   -
COLDER   661   -
CONV3C  1442   -
CONV3D  1465   -
COPYBF   306   -
CSERE0   400   -
CSEREX   376   -
CSNOFD   364   -
CSRDY    341   -
DATALL  1301   -
DATSER  1206   -
DATSUB  1166   -
DFBDCK  1140   -
DSPERR   405   -
FINDID     6   -
FNID10    13   -
FNID60    62   -
IFC      113   -
INTCAL  1423   -
INTDIR   361   -
PCHKKB  1470   -
PCKBRT  1512   -
PURGEF   135   -
PURGEP   140   -
PWRDN   1712   -
READA    606   -
READK   1001   -
REQADR  1100   -
REWENT   145   -
RSTBP   1644   -
SEEK    1565   -
SEEK40  1640   -
SEEKC   1564   -
SEEKN   1562   -
SEEKR2  1557   -
SEEKRC  1571   -
SEEKRD  1572   -
SEEKRN  1567   -
SETBP0  1626   -
SETBPC  1645   -
SETBPL  1627   -
SETBPT  1631   -
SNBYTS  1674   -
STOPIO   111   -
SVBREN  1317   -
SVENTR  1363   -
SVENTW  1365   -
```

```
SVMODE  1417   -
VERIFY   443   -
WRET10   160   -
WRET15   171   -
WRTA     467   -
WRTA20   571   -
WRTK     704   -
```

EXTERNAL REFERENCES (SCPL4B - ILCAS&CTL QUAD 3 = CS3 = ADR @76000-77777)

ACKX	1465	
ACKX	1466	
AD2-10	1434	
AD2-10	1435	
AOUT1	11	
AOUT1	12	
ASN15	1057	
ASN15	1060	
BCDBIN	1175	1230
BCDBIN	1176	1231
BINBDC	63	
BINBDC	64	
CHKLAD	1746	
CHKLAD	1747	
CHKPCT	140	
CHKPCT	141	
CKSMER	661	
CKSMER	662	
COLDST	663	
COLDST	664	
CRTFL	545	
CRTFL	546	
CRTFL0	744	
CRTFL0	745	
CSEOF	1271	
CSEOF	1272	
CSEREX	772	
CSEREX	773	
CSERR	1622	
CSERR	1623	
CSSTAS	343	1613
CSSTAS	344	1614
DATSER	1315	
DATSER	1316	
DTFLOW	173	
DTFLOW	174	
ERR110	403	
ERR110	404	
ERRDE	1461	
ERRDE	1462	
ERRNE	1210	
ERRNE	1211	
ERRPR	503	
ERRPR	504	
FLINKA	500	
FLINKA	501	
FLSCH	511	705
FLSCH	512	706
FLSCH0	610	1003
FLSCH0	611	1004
FLSCHI	443	
FLSCHI	444	
FLSCHJ	135	
FLSCHJ	136	
FLTYER	1156	1351
FLTYER	1157	1352

```
SNDRDN   577
SNDRDN   600
SNDRGA   557   572
SNDRGA   560   573
SNDRGC   575
SNDRGC   576
SRWRT    162
SRWRT    163
TALKER    13  1100
TALKER    14  1101
UNL      333  1206
UNL      334  1207
UNT      455   653
UNT      456   654
WAITS    302
WAITS    303
WKUP80   674
WKUP80   675
WKUPLP  1736
WKUPLP  1737
WPROM    470
WPROM    471
WRLPBK   322  1402
WRLPBK   323  1403
WRTA20   757
WRTA20   760

End of VASM assembly
```