intel

APPLICATION NOTE

AP-80

January 1980

ISIS-II System Calls

Sherri Gavarre
Microcomputer Development Systems Applications

121559-001 Rev. A

RELATED INTEL PUBLICATIONS

ISIS-II User's Guide, 9800306
8080/8085 Assembly Language Programming Manual, 9800301
PL/M Programming Manual, 9800268

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

ISIS-II System Calls

Table of Contents

I. INTRODUCTION

The Intel Systems Implementation Supervisor (ISIS-II) provides the user with the capability to call ISIS-II routines, or procedures, (referred to in the ISIS manual as "System Calls") in the user's program. The ISIS-II routines are compiled or assembled procedures which reside within the operating system. These routines are used by many of the ISIS utilities to perform some of the common tasks such as input and output for a program. The use of ISIS routines facilitates programming as they perform many necessary system interface functions, and are invoked by only an ISIS call instruction to the desired routine.

The most used functions that are available through ISIS system calls are interface routines to perform input and output for a specified device, routines to perform ISIS diskette directory maintenance, and routines that handle program loading and execution.

This note covers in detail the use of system calls for input and output operations (I/O). It describes the uses for ISIS input and output routines, how to access these routines from programs in PL/M-80 and ASM-80, ISIS calls used in I/O, and line edited files. Any ISIS system calls not discussed here follow the same format, and can be used in the same manner.

II. THE USE OF ISIS ROUTINES FOR I/O

The ISIS I/O routines direct the I/O operations to and from "files" that are defined in the system configuration. An ISIS "file" is an ISIS disk file, or any other standard Intellec peripheral in the system configuration (i.e., CRT, Line Printer, etc.). Therefore, the ISIS I/O routines allow the program to easily communicate with the system's peripherals, referred to as 'files' in this application.

An input operation from an ISIS file is accomplished by using the ISIS READ routine. The READ routine transfers a specified number of bytes from an ISIS file to system memory. An output operation to an ISIS file is done by using the ISIS WRITE routine. The WRITE routine transfers a specified number of bytes from system memory to the file. (See figure 1.)

Figure 1

121559-1

ISIS associates the logical stream of data, for input and output system routines, with the physical file through the use of active file numbers. An active file number (AFTN) is assigned to each physical file that is used for input or output by the program and serves as a connection between ISIS and the physical file. (See figure 2.)

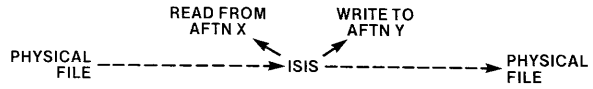

Figure 2

121559-2

In order for ISIS to assign an AFTN to a physical file, the user must specify to ISIS that the program will be using that physical file for input and output operations. This is done by using the system routine OPEN. During the OPEN system routine, an AFTN will be assigned to the physical file and returned to the calling program. The AFTN will then be used by the user's program to identify the file to ISIS in later system routines.

OPEN :LP: -----> AFTN = 4
(In ISIS AFTN table and returned to program)
WRITE TO AFTN 4 -----> ISIS AFTN TABLE -----> :LP:

The AFTN is assigned until the file is closed, by the close routine, or the user's program is terminated and an exit routine called. ISIS allows only six files to be open simultaneously. If the program will be communicating with more than six files, some files will have to be closed so that others can be opened.

III. ACCESS OF ISIS ROUTINES FROM THE USER'S PROGRAM

The ISIS routines are accessed from the user's program through a call instruction in either PL/M-80 or ASM-80 code. Since the addresses of the ISIS routines are outside of the user's program, the ISIS routines must be defined as external. The external references (calls to the ISIS routines) must be resolved before the program is executed so that the call instructions have the correct address of the ISIS routine. This is done by linking the assembled or compiled program to SYSTEM.LIB. SYSTEM.LIB is an ISIS library that contains routines which interface the user's program with the ISIS system. (See figure 3.)

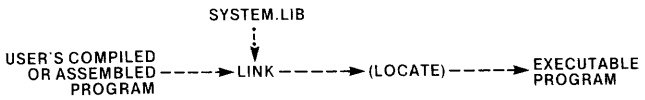

Figure 3 121559-3

The following commands are an example of the procedure outlined in figure 3.

```
-ASM80 VIEW.SRC <CR>  (Assemble the program)
-LINK VIEW.OBJ, SYSTEM.LIB TO VIEW.LNK
     (Resolves external references)
-LOCATE VIEW.LNK
```

The output of locate will be VIEW, the executable file.

The link operation with SYSTEM.LIB will insert the correct entry point of the ISIS routine into the call instruction. Therefore, at execution time the call instructions will cause a branch into the ISIS routine. When the execution of the routines is completed, control returns to the user's program with any results in the specified parameters. (See figure 4.)

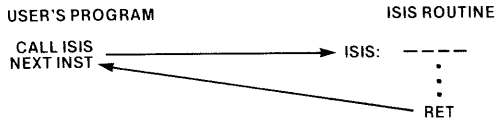

Figure 4 121559-4

IV. PASSING PARAMETERS TO THE ISIS ROUTINES

In order for the ISIS routine to perform a desired function on a certain file, the user's program must send information to the ISIS routine in the form of parameters. The parameters define the file that is to be used and supply other necessary information such as the number of bytes to transfer.

For each routine, a given number of parameters must be defined and passed to specify the nature of the operation that the routine is to perform. Parameter passing differs in assembly language and PL/M, but the actual parameters being passed are the same. Therefore, PL/M-80 and ASM-80 calls will be discussed separately.

ASM80 Calls to the ISIS Routines

The user's ASM-80 program and the ISIS routines must communicate information to each other during the execution of the program. An area of communication, defined in the user's program, is necessary for the passing of information. This area is defined as the system call parameter block. The user's program places information for the ISIS routine in the parameter block, and the ISIS routine returns information to the calling program in the block. (See figure 5.) Therefore, there is a parameter block format for each call so the system routine can obtain the needed information.

Figure 5 121559-5

The entry point into ISIS for ASM-80 calls to system routines is labeled ISIS and all calls are made to this address. A routine identifier must be sent to ISIS to identify the correct routine to execute. The routine identifiers for I/O operations are defined by ISIS and listed below:

OPEN	0
CLOSE	1
READ	3
WRITE	4
ERROR	12
EXIT	9

There are two parameters sent to ISIS:

1. The number that identifies the ISIS routine
2. The address of the start of the parameter block

The first parameter, the routine identifier, is sent to Register C. The address of the parameter block is sent in Register DE.

The instructions to load the parameters would be:

```
MVI   C,ROUTINE IDENTIFIER
LXI   D,PARMBLOCK
```

The parameter block is defined in the user's program as a contiguous area of storage. Each parameter has one word of storage allocated for it.

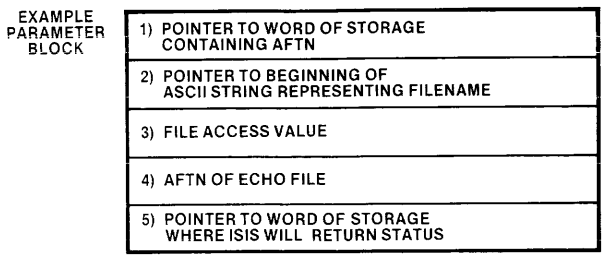

Figure 6 121559-6

Figure 6 shows the parameter block that is defined by the open routine. (See open routine definition for further explanation.) Each element represents a parameter and a word of storage. This block could be coded in the user's program as:

```
OPEN:  DW OAFTN  ;POINTER TO AFTN
       DW OFILE  ;POINTER TO FILENAME
       DW 1      ;ACCESS VALUE
       DW 0      ;ECHO FILE AFTN
       DW OSTAT  ;POINTER TO OSTAT
```

This represents five words of storage, the parameter block. Three types of parameters are represented here:

1. Numerical Parameters
2. String Parameters
3. Return Parameters

Numerical parameters are passed by value. That is, the parameter to be passed is passed directly, as in the access value, and the echo file AFTN parameters above.

String parameters are passed by address (pointer). That is, the address of the ASCII string is passed in the parameter block. In the following example, OFILE is in the parameter block and is a pointer, or address, to the ASCII string that identifies the file:

```
        DW OFILE              ;POINTER TO FILENAME
        (THIS DW IS LOCATED IN PARAMETER BLOCK)
        •
        •
        •
OFILE:  DB ':F1:EXAMPL.FIL' ;FILENAME
```

Return parameters are pointers to variables in which the ISIS routine will return a value. This parameter is similar to the string parameter except that data is not stored in the variable that is pointed to in the parameter block:

```
        DW OSTAT ;POINTER TO STATUS VARIABLE
        •  (THIS DW IS LOCATED IN PARAMETER
           BLOCK)
        •
        •
        •
OSTAT:  DS 2          ;ONE WORD OF STORAGE WHERE
                      ;ISIS WILL RETURN THE STATUS
```

PL/M-80 Calls to ISIS Routines

The parameters that are sent to an ISIS routine from a PL/M-80 program are the same as the parameters sent from an assembly language program, but the call to an ISIS routine is different.

Calling ISIS routines from PL/M-80 is similar to calling any procedure in PL/M-80. A call is made directly to the name of the ISIS routine with the list of parameters preceeded by the call statement.

Example:

```
CALL OPEN (.AFTN,.FILE,1,0,.STATUS);
```

Since the ISIS routines are in SYSTEM.LIB, they must be delcared as an external procedure to the user's program.

```
OPEN: PROCEDURE (AFTNPTR, FILE,
ACCESS, MODE, STATUS) EXTERNAL;
DECLARE (AFTNPTR, FILE, ACCESS, MODE,
STATUS) ADDRESS;
END OPEN;
```

In the example call instruction, the list of parameters is equivalent to the parameter block in the ASM-80 example. (See figure 6.)

There are five parameters for the open routine, each of word length. One of the parameters, FILE, is a string parameter, (see figure 6), which is the address of the location where an ASCII string is stored. In PL/M-80, the string parameters are defined in the call parameter list with the dot (.) operator, since the address is being passed as a parameter. Two of the parameters, AFTN and STATUS are return parameters, which are addresses of variables where ISIS will return a value.

These variables must be defined in the user's program:

```
DECLARE FILE (10) BYTE DATA
('':F1:EXAMPL')

DECLARE AFTN ADDRESS
DECLARE STATUS ADDRESS;
```

The other parameters, ACCESS and MODE, are numerical parameters and are passed by value in the call routine instruction.

V. THE ISIS ROUTINES

In this section, each of the ISIS I/O routines is described in detail with references to the example program VIEW which follows the routine descriptions.

OPEN Routine

The OPEN routine connects a file to be used for input or output to the ISIS system by assigning an active file number and allocating buffers for the I/O operations. The active file number (AFTN) will be used by the user's program and ISIS to identify this file in all further operations. Every file that will be used in a system routine must be opened first through the use of this routine. There are two exceptions, :CI: and :CO: are always open.

The user calls the OPEN routine with a pointer to the name of the file that is to be opened, and ISIS returns the active file number in the two byte area pointed to by the AFTN parameter. ISIS will also return an error status if any exists.

Each parameter is a word of storage and is either an address of the actual parameter or an absolute value. (See figure 7.) An example of an open parameter block can be found in the example program, lines 38-43.

The parameters are as follows:

1. The address of a word of storage where ISIS will return the AFTN that has been assigned to the file.

2. The address of the character string that is the name of the file to be opened. The filename string must be defined in the user's program.

3. The mode of access of the file:
   - =1 This file will be read from
   - =2 This file will be written to
   - =3 This file will be read and written to

4. The AFTN of the echo file. This parameter is used only for line edited files.

5. The address of a word of storage where ISIS will return any errors.

Parameter 2 is a string parameter, a pointer to the filename STRING. In the example, the filename STRING is in RBUF as it was read from the console. Parameters 1 and 5 are return parameters, pointers to the two one-word storage locations where ISIS will return the values. Parameters 3 and 4 are numerical parameters, and their value is in the parameter block, lines 41 and 42.

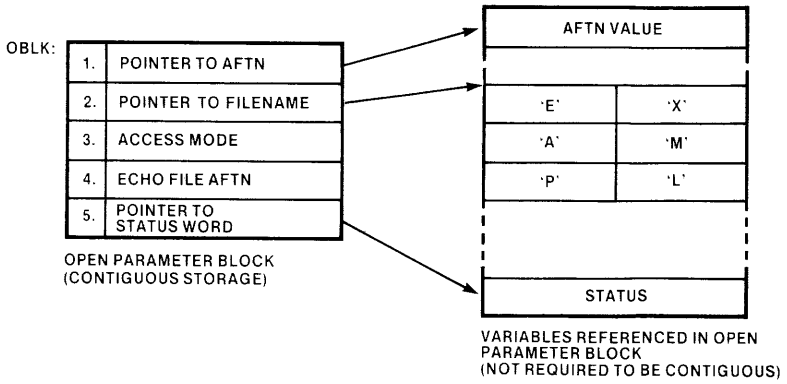

Figure 7. Open Routine Parameter Block with Pointer Referenced Variables 121559-7

READ Routine

The READ routine transfers a specified number of bytes from a file to a buffer area in memory. The buffer area can be in the user's program or can be an area of user memory. Caution must be taken to see that the buffer area is in available space and that nothing will be destroyed. If the buffer area is to be in the user's program, it is defined through the define storage directive (DS).

Five parameters are required for the READ routine. Each parameter is of word length. The READ routine has two return parameters, the actual number of characters read, and any error status. (See figure 8.)

The file that will be read from is identified through the active file number (AFTN) that was returned by the OPEN routine. The AFTN is sent to the READ routine as another parameter.

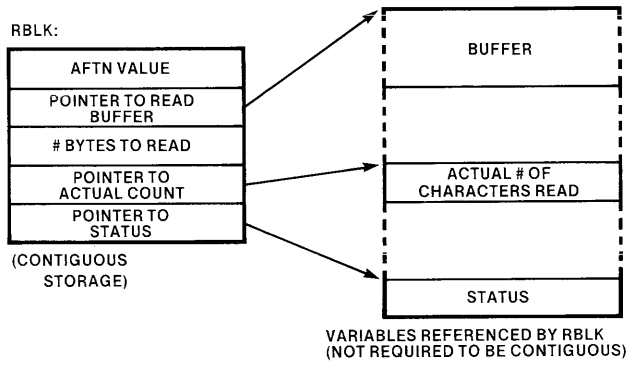

Figure 8. Read Routine Parameter Block 121559-8

The parameters are as follows:

1. The active file number (AFTN) of the file that is to be read.
2. The starting address of the memory buffer where the data will be stored.
3. The number of bytes that are to be read from the file.
4. The address of the memory location where the READ routine will return the number of bytes actually read.
5. The address of a word of storage where ISIS can return any errors.

An example of a READ parameter block can be found in lines 63-68 of 'VIEW'. Parameter 2 is a string parameter, a pointer to 'RBUF' that is defined in line 106. Parameters 4 and 5 are return parameters, pointers to 'ACT' where ISIS will return value (line 74), and 'STAT' where ISIS will return any error (line 95).

WRITE Routine

The ISIS WRITE routine transfers data from a buffer in memory to a user specified file. The buffer is an area of storage as in the READ routine.

The data in the buffer could have been defined by a program, or left from an operation, such as an ISIS READ routine.

The WRITE routine requires four parameters of word length and returns one parameter, the error status. (See figure 9.)

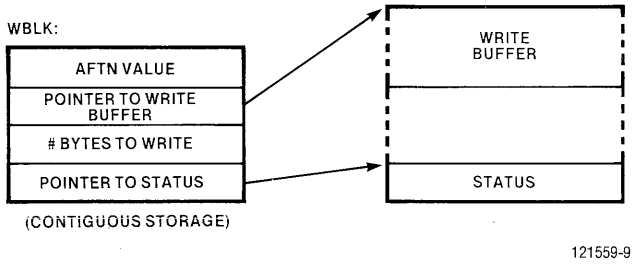

Figure 9. Write Routine Parameter Block with Pointer Referenced Variables

The parameters are as follows:

1. The active file number (AFTN) of the file to which the data buffer will be written. This file must have already been opened by the ISIS OPEN routine.
2. The address of the buffer where the data to output is stored.
3. The number of bytes to transfer from the buffer to the file.
4. The address of the memory location where ISIS will return the errors.

An example of a WRITE parameter block can be found in lines 79-83.

Parameter 2 is a string parameter pointing to the buffer. Parameters 1 and 3 are numerical parameters.

ERROR Routine

The ERROR routine enables the user's program to send an erorr message to the console.

This routine is useful for applications programs where the user has defined error numbers that are to be sent to the console. It is also used in programs that use the ISIS system calls, sending to the console any error messages that are returned from ISIS routines.

In the example program VIEW, all of the ISIS routine parameter blocks point to the first parameter of the error block for the status parameter. (See figure 10.)

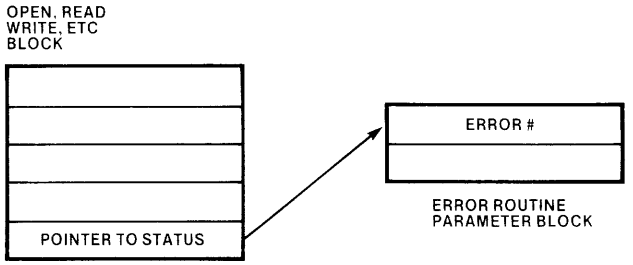

Figure 10

121559-10

In this way if ISIS returns an error, it would return it to this parameter block, saving a MOVE instruction.

The error parameter block consists of two parameters for assembly language programs, and one parameter for PL/M programs. The parameters are as follows: (line 95)

1. The error number to output to the console
2. Assembly language only: The address of a memory location where ISIS can return any error

CLOSE File Routine

The ISIS CLOSE routine releases the file's AFTN and the buffers allocated for I/O. When a file is no longer being used for I/O, it should be closed.

Two parameters are required for the close parameter block:

1. The AFTN of the file to be closed
2. The address of a word of storage where ISIS will return any error condition

The CLOSE parameter block can be found in lines 89 to 90.

EXIT Routine

The EXIT routine terminates execution of the user's program and returns control to ISIS. An EXIT call, or a similar routine is necessary to return control to ISIS.

The EXIT routine closes all of the open files except :CI: and :CO:. Only one parameter is necessary:

1. The address of the memory location where ISIS will return any error conditions. (line 100)

VI. EXAMPLE PROGRAM

```
ASM80 VIEW2.SRC MACROFILE


ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    1

  LOC  OBJ          LINE        SOURCE STATEMENT

                       1
                       2 ;VIEW IS AN ISIS COMMAND OF THE FORM:-
                       3 ;-VIEW FILENAME
                       4 ;
                       5 ; IT DISPLAYS THE CONTENTS OF THE FILE IN HEX AND (IF POSSIBLE)
                       6 ; IN ASCII. EACH LINE OF THE DISPLAY SHOWS 16 BYTES AND THE
                       7 ; DISPLAY IS HALTED AFTER EACH BLOCK OF 16 LINES, AND CAN BE
                       8 ; RESTARTED BY A <RETURN> ENTERED AT THE KEYBOARD.  IF <ESC>
                       9 ; IS ENTERED, VIEW RETURNS TO ISIS.
                      10 ;
                      11 ;
                      12 ;  EXTERNAL DIRECTIVE FOR LINK TO ISIS ROUTINES
                      13 ;
                      14          EXTRN   ISIS
                      15
                      16          CSEG
                      17 ;
                      18 ;   EQUATE DIRECTIVES FOR ROUTINE IDENTIFIERS
                      19 ;
                      20 ;
  0000                21 OPEN     EQU     0
  0001                22 CLOSE    EQU     1
  0003                23 READ     EQU     3
  0004                24 WRITE    EQU     4
  0009                25 EXT      EQU     9
  000C                26 ERROR    EQU     12
                      27 ;
                      28 ;
                      29 ;
                      30 ;  OPEN ROUTINE PARAMETER BLOCK FOR FILE
                      31 ;     TO BE VIEWED
                      32 ;    -'FILENAME' IS READ INTO RBUF BY A PRECEEDING
                      33 ;       READ CALL
                      34 ;    - THE AFTN IS RETURNED TO 'RBLK', A PARAMETER IN
                      35 ;      THE READ PARAMETER BLOCK.  THIS IDENTIFIES THE
                      36 ;      FILE OPENED HERE FOR THE READ ROUTINE.
                      37 ;
  0000 1400      C    38 ORBLK:   DW      RBLK            ; ADDRESS OF VARIABLE WHERE
                      39                                  ; AFTN WILL BE STORED BY ISIS
  0002 3000      C    40          DW      RBUF            ; ADDRESS OF 'FILENAME' STRING
  0004 0100           41          DW      1               ; ACCESS MODE FOR READ
  0006 0000           42          DW      0               ; ECHO FILE (NOT USED HERE)
  0008 2C00      C    43          DW      STAT            ; ADDRESS OF VARIABLE WHERE
                      44                                  ; ISIS RETURNS STATUS ; (LOCATED
                      45                                  ; IN ERROR PARAMETER BLOCK)
                      46 ;
                      47 ;
                      48 ; READ PARAMETER BLOCK FOR READ 'FILENAME' FROM CONSOLE
                      49 ;
  000A 0100           50 RCON:    DW      1               ; AFTN OF :CI:
  000C 3000      C    51          DW      RBUF            ; ADDRESS OF BUFFER WHERE
                      52                                  ; FILENAME IS STORED
  000E 2000           53          DW       32             ; # OF BYTES TO READ
  0010 1E00      C    54          DW      ACT             ; # OF BYTES READ INTO BUFFER
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    2

  LOC  OBJ          LINE        SOURCE STATEMENT

 0012 2C00      C     55          DW      STAT            ; ADDRESS OF STATUS VARIABLE
                      56                                  ; (LOCATED IN ERROR BLOCK)
                      57 ;
                      58 ;
                      59 ;  READ ROUTINE PARAMETER BLOCK FOR READ FROM FILE OPENED
                      60 ;   FOR VIEW
                      61 ;   - AFTN IS IN RBLK; WAS PLACED BY THE OPEN ROUTINE
                      62 ;
 0014                 63 RBLK:   DS      2               ; THE AFTN OF FILE TO BE READ FROM
 0016 3000      C     64         DW      RBUF            ; ADDRESS OF BUFFER FOR STORE
 0018 0001            65         DW      256             ; # OF BYTES TO READ
 001A 1E00      C     66         DW      ACT             ; # OF BYTES READ ( RETURNED
                      67                                 ;  BY ISIS)
 001C 2C00      C     68         DW      STAT            ; ADDRESS OF STATUS VARIABLE
                      69                                 ; (LOCATED IN ERROR BLOCK)
                      70 ;
                      71 ; VARIABLE DEFINED IN READ BLOCKS WHERE ISIS WILL RETURN
                      72 ;  ACTUAL NUMBER OF BYTES READ
                      73 ;
 001E                 74 ACT:    DS      2
                      75 ;
                      76 ;
                      77 ;  WRITE ROUTINE PARAMETER BLOCK (WRITE HEX FILE TO CONSOLE)
                      78 ;
 0020 0000            79 WBLK:   DW      0               ; AFTN OF :CO:
 0022 3001      C     80         DW      WBUF            ; ADDRESS OF BUFFER THAT CONTAINS
                      81                                 ; DATA TO OUTPUT
 0024 4600            82 WCNT:   DW      70              ; # OF BYTES TO WRITE TO CONSOLE
 0026 2C00      C     83         DW      STAT            ; ADDRESS OF ERROR STATUS
                      84 ;
                      85 ;
                      86 ;  CLOSE ROUTINE PARAMETER BLOCK
                      87 ;    - AFTN MOVED TO CBLK DURING PROGRAM EXECUTION
                      88 ;
 0028                 89 CBLK:   DS      2               ; AFTN OF FILE TO BE CLOSED
 002A 2C00      C     90         DW      STAT            ; ADDRESS OF ERROR STATUS
                      91 ;
                      92 ;
                      93 ;  ERROR ROUTINE PARAMETER BLOCK
                      94 ;
 002C                 95 STAT:   DS      2               ; ERROR STATUS FOR EACH ROUTINE IS
                      96                                 ;  RETURNED HERE
                      97 ;
                      98 ;  EXIT ROUTINE PARAMETER BLOCK
                      99 ;
 002E 2C00      C    100 XBLK:   DW      STAT            ; ERROR STATUS RETURNED HERE
                     101 ;
                     102 ;
                     103 ;  RBUF - BYTES READ FROM THE VIEW FILE ARE STORED IN RBUF
                     104 ;  WBUF - HEX CONVERTED DATA IS STORED IN WBUF UNTIL OUTPUT
                     105 ;
 0030                106 RBUF:   DS      256
 0130                107 WBUF:   DS       70
                     108 ;
                     109 ;
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    3


 LOC  OBJ          LINE        SOURCE STATEMENT

                    110 ;
 0176 310000    S   111 START:  LXI     SP,STACK        ; INITIALIZE STACK POINTER
                    112 ;
                    113 ;  CALL READ ROUTINE TO READ INTO RBUF THE FILENAME INPUT
                    114 ;   BY THE USER
                    115 ;
 0179 0E03          116         MVI     C,READ
 017B 110A00    C   117         LXI     D,RCON
 017E CD3102    C   118         CALL    ISIS0
                    119 ;
                    120 ;  CALL OPEN ROUTINE TO OPEN FILENAME THAT WAS INPUT BY USER
 0181 0E00          121         MVI     C,OPEN
 0183 110000    C   122         LXI     D,OBLK
 0186 CD3102    C   123         CALL    ISIS0
                    124 ;
                    125 ; BEGIN LOOP TO READ 256 BYTES AT A TIME INTO BUFFER
                    126 ;  FOR CONVERSION TO HEX AND ASCII.  THE BYTES ARE
                    127 ;  CONVERTED AND STORED IN WBUF UNTIL 16 HAVE BEEN
                    128 ;  CONVERTED; THE BUFFER IS OUTPUT AND THE INNER
                    129 ;  LOOP CONTINUES UNTIL THE READ BUFFER IS EXHAUSTED.
                    130 ;  ANOTHER 256 BYTES ARE READ, UNTIL FILE IS DONE.
                    131 ;
                    132 ;
 0189 0E03          133 LOOP:   MVI     C,READ          ;READ 256 BYTES TO RBUF
 018B 111400    C   134         LXI     D,RBLK
 018E CD3102    C   135         CALL    ISIS0
                    136 ;
                    137 ;  IF ACTUAL BYTES READ IS ZERO, EOF ENCOUNTERED: EXIT
                    138 ;
 0191 2A1E00    C   139         LHLD    ACT
 0194 7D            140         MOV     A,L
 0195 B4            141         ORA     H
 0196 CA1102    C   142         JZ      DONE2
                    143 ;
                    144 ;  OTHERWISE, STORE ACTUAL # OF BYTES READ IN E
                    145 ;
 0199 5D            146         MOV     E,L
                    147 ;
 019A 213000    C   148         LXI     H,RBUF  ;POINT TO BUFFER
                    149 ;
                    150 ;  NEWLIN IS LOOP FOR CODE TO WRITE ONE LINE TO CONSOLE
                    151 ;
                    152 ;
 019D 013001    C   153 NEWLIN: LXI     B,WBUF  ; POINTER TO WRITE BUFFER
 01A0 CD5402    C   154         CALL    CRLF    ; ROUTINE TO STORE CR-LF
 01A3 160F          155         MVI     D,15    ; STORE COUNT FOR 1 LINE
 01A5 D5            156         PUSH    D       ; SAVE LINE COUNT AND ADDRESSES
 01A6 E5            157         PUSH    H       ;  ON STACK
                    158 ;
                    159 ;
                    160 ;
 01A7 7E            161 LINE:   MOV     A,M     ; MOVE BYTE INTO A
 01A8 0F            162         RRC             ; ROTATE FOR HIGH NIBBLE
 01A9 0F            163         RRC
 01AA 0F            164         RRC
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    4

  LOC  OBJ          LINE        SOURCE STATEMENT

  01AB 0F            165          RRC
  01AC CD4902    C   166          CALL    LOADBF  ; LOAD CHARS INTO WRITE BUFFER
  01AF 7E            167          MOV     A,M     ; MOVE SAME BYTE INTO A
  01B0 CD4902    C   168          CALL    LOADBF
  01B3 3E20          169          MVI     A,' '   ; STORE SPACE IN WRITE BUFFER
  01B5 02            170          STAX    B       ; TO SEPARATE HEX
  01B6 03            171          INX     B       ;
  01B7 23            172          INX     H       ; POINT TO NEXT CHARACTER IN RBUF
  01B8 1D            173          DCR     E       ; DECREMENT READ BUFFER COUNT
  01B9 CAC001    C   174          JZ      EOB     ;END OF INPUT BUFFER?
  01BC 15            175          DCR     D       ; DECREMENT LINE COUNT
  01BD F2A701    C   176          JP      LINE    ;LOOP 16 TIMES
                     177 ;
                     178 ;
                     179 ;  END OF LINE:  STORE SPACES TO SEPARATE HEX FROM ASCII
                     180 ;
  01C0 02            181 EOB:     STAX    B
  01C1 03            182          INX     B
  01C2 02            183          STAX    B
  01C3 03            184          INX     B
  01C4 02            185          STAX    B
  01C5 03            186          INX     B
  01C6 15            187          DCR     D       ;CHARACTER LINE COUNT
  01C7 F2C001    C   188          JP      EOB     ;LOOP TO EQUALIZE NO. OF CHARS.
  01CA E1            189          POP     H       ; POP INPUT BUFFER POINTER
  01CB D1            190          POP     D       ; POP LINE COUNT
                     191 ;
                     192 ;  CONVERT BYTES TO ASCII
                     193 ;
  01CC 7E            194 ASCII:   MOV     A,M
  01CD E67F          195          ANI     7FH
  01CF FE20          196          CPI     ' '
  01D1 D2D601    C   197          JNC     ALFA?
  01D4 3E20          198          MVI     A,' '
  01D6 FE60          199 ALFA?:   CPI     60H
  01D8 DADD01    C   200          JC      ALNUM    ;ONLY PRINTS UPPER CASE ASCII
  01DB 3E20          201          MVI     A,' '
  01DD 02            202 ALNUM:   STAX    B       ; STORE CHARACTER IN BUFFER
  01DE 03            203          INX     B
  01DF 23            204          INX     H
  01E0 1D            205          DCR     E
  01E1 C2EC01    C   206          JNZ     NXCHAR
                     207 ;
                     208 ;  FILL BUFFER WITH SPACES IF LESS THAN 16 BYTES
                     209 ;
  01E4 3E20          210 FILBUF:  MVI     A,' '
  01E6 02            211          STAX    B       ;STORE BLANK
  01E7 03            212          INX     B
  01E8 15            213          DCR     D
  01E9 F2E401    C   214          JP      FILBUF
                     215 ;
  01EC 15            216 NXCHAR:  DCR     D
  01ED F2CC01    C   217          JP      ASCII   ;LOOP TO PRINT 16 CHARS
  01F0 CD5D02    C   218          CALL    WRTBUF  ;CALL ROUTINE TO WRITE BUFFER
  01F3 1C            219          INR     E
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    5


 LOC  OBJ          LINE        SOURCE STATEMENT

 01F4 1D            220          DCR     E
 01F5 C29D01    C   221          JNZ     NEWLIN  ;NEW LINE UNTIL BUFFER EXHAUSTED
                    222 ;
                    223 ;  END OF INPUT BUFFER:  IF LESS THAN 256 CHARS READ,
                    224 ;    ALL OF FILE HAS BEEN READ, DONE2.  OTHERWISE
                    225 ;    WAIT FOR AN INPUT OF <CR> OR <ESC> FROM THE CONSOLE
                    226 ;    TO CONTINUE (LOOP) OR EXIT.
                    227 ;
 01F8 3A1E00    C   228 DONE:   LDA     ACT     ;SEE IF LESS THAN 256 CHARS
 01FB B7            229          ORA     A
 01FC C21102    C   230          JNZ     DONE2   ;MUST BE END OF FILE IF LESS THAN 256 CHARS
 01FF 0E03          231          MVI     C,READ  ; READ INPUT FROM CONSOLE: <CR> OR <ESC>
 0201 110A00    C   232          LXI     D,RCON
 0204 CD3102    C   233          CALL    ISIS0
 0207 3A3000    C   234          LDA     RBUF
 020A E67F          235          ANI     7FH
 020C FE1B          236          CPI     1BH     ; IF INPUT = <ESC> , EXIT
 020E C28901    C   237          JNZ     LOOP
                    238 ;
                    239 ;  FILE HAS BEEN READ AND CONVERTED
                    240 ;   WRITE FINAL CR-LF TO CONSOLE
                    241 ;
 0211 210200        242 DONE2:  LXI     H,2     ; CHANGE # OF BYTES TO WRITE
 0214 222400    C   243          SHLD    WCNT    ; IN WRITE PARAMETER BLOCK
 0217 013001    C   244          LXI     B,WBUF
 021A CD5402    C   245          CALL    CRLF    ; INSERT CR-LF IN WBUF
 021D CD5D02    C   246          CALL    WRTBUF  ; OUTPUT BUFFER
                    247 ;
                    248 ;  MOVE READ FILE AFTN TO CLOSE ROUTINE PARAMTER BLOCK
                    249 ;
 0220 2A1400    C   250          LHLD    RBLK    ;FETCH FILE AFTN
 0223 222800    C   251          SHLD    CBLK
                    252 ;
 0226 0E01          253          MVI     C,CLOSE   ; CLOSE FILE
 0228 112800    C   254          LXI     D,CBLK
 022B CD3102    C   255          CALL    ISIS0
                    256 ;
 022E C34102    C   257          JMP     EXIT
                    258 ;
                    259 ;
                    260 ;  ISIS0 - CALLS ISIS AND TESTS ERROR STATUS UPON RETURN FROM
                    261 ;           ISIS
                    262 ;
 0231 CD0000    E   263 ISIS0:  CALL    ISIS
 0234 3A2C00    C   264          LDA     STAT
 0237 B7            265          ORA     A
 0238 C8            266          RZ
                    267 ;
                    268 ;    IF NON-ZERO DROP THROUGH TO CALL TO ERROR ROUTINE
                    269 ;
 0239 0E0C          270          MVI     C,ERROR
 023B 112C00    C   271          LXI     D,STAT
 023E CD0000    E   272          CALL    ISIS
                    273 ;
                    274 ;...AND TO EXIT.
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    6

  LOC  OBJ          LINE        SOURCE STATEMENT

                     275 ;
  0241 0E09          276 EXIT:   MVI     C,EXT     ;EXIT ROUTINE IDENTIFIER
  0243 112E00    C   277         LXI     D,XBLK
  0246 CD0000    E   278         CALL    ISIS
                     279 ;
                     280 ;  EXIT RETURNS CONTROL TO ISIS; END OF PROGRAM
                     281 ;
                     282 ;
                     283 ;  LOADBUF SUBROUTINE- CONVERTS A NIBBLE TO HEX AND
                     284 ;   LOADS INTO THE WRITE BUFFER
                     285 ;
                     286 ;
  0249 E60F          287 LOADBF: ANI     0FH
  024B C690          288         ADI     90H
  024D 27            289         DAA
  024E CE40          290         ACI     40H
  0250 27            291         DAA
  0251 02            292         STAX    B               ; STORE IN WBUF
  0252 03            293         INX     B
  0253 C9            294         RET
                     295 ;
                     296 ;
                     297 ;  CRLF - STORES A CARRIAGE RETURN AND LINE FEED INTO WBUF
                     298 ;
  0254 3E0D          299 CRLF:   MVI     A,0DH
  0256 02            300         STAX    B
  0257 03            301         INX     B
  0258 3E0A          302         MVI     A,0AH
  025A 02            303         STAX    B
  025B 03            304         INX     B
  025C C9            305         RET
                     306 ;
                     307 ;
                     308 ;  WRTBUF - CALLS THE WRITE ROUTINE AND OUTPUTS WBUF
                     309 ;           TO THE CONSOLE
                     310 ;
  025D D5            311 WRTBUF: PUSH    D
  025E E5            312         PUSH    H
  025F 0E04          313         MVI     C,WRITE
  0261 112000    C   314         LXI     D,WBLK
  0264 CD3102    C   315         CALL    ISIS0
  0267 E1            316         POP     H
  0268 D1            317         POP     D
  0269 C9            318         RET
                     319 ;
                     320 ;
  0176           C   321         END     START

PUBLIC SYMBOLS

EXTERNAL SYMBOLS
ISIS    E 0000
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE   PAGE    7


USER SYMBOLS
ACT     C 001F    ALFA?   C 01D6    ALNUM   C 01DD    ASCII   C 01CC    CBLK    C 0028    CLOSE   A 0001    CRLF    C 0254
DONE    C 01F8    DONE2   C 0211    EOB     C 01C0    ERROR   A 000C    EXIT    C 0241    EXT     A 0009    FILBUF  C 01E4
ISIS    E 0000    ISIS0   C 0231    LINE    C 01A7    LOADBF  C 0249    LOOP    C 0189    NEWLIN  C 019D    NXCHAR  C 01EC
OBLK    C 0000    OPEN    A 0000    RBLK    C 0014    RBUF    C 0030    RCON    C 000A    READ    A 0003    START   C 0176
STAT    C 002C    WBLK    C 0020    WBUF    C 0130    WCNT    C 0024    WRITE   A 0004    WRTBUF  C 025D    XBLK    C 002E

ASSEMBLY COMPLETE,   NO ERRORS
```

VII. LINE EDITED FILES

Line editing provides a way for the user's program to read ASCII files from a file on a line by line basis. Instead of transmitting the data from the file directly to the user buffer, ISIS stores the data in an intermediate buffer, the line edit buffer. The data in the line edit buffer can be edited before it is sent to the user's buffer. This provides a chance for correction of errors made in input.

The most common use of line editing is in reading from a keyboard file. The user can correct any errors in the current line that is being entered before it is transmitted to the program buffer. (See figure 11.)

Figure 11

121559-11

Line editing is initiated when a file, opened for line editing, is read by the ISIS READ routine. When the READ is executed, input is accepted into the line edit buffer from the line edited file.

Corrections are made to the data in the line edit file during the READ by using the line editing control characters:

RUBOUT-Deletes character preceeding cursor from line edit buffer

CTRL/P-Literalizes next character

CTRL/R-Echos current contents of line edit buffer

CTRL/X-Deletes entire contents of line edit buffer

CTRL/Z-End of file character

When one of these characters is entered at the line edit file, the control function is performed without storing the control character in the line edit buffer.

Input into the line edit buffer is terminated when any of the following occurs:

1. A carriage return or line feed is entered from the input file.
2. The line edit buffer is filled (122 characters).
3. An escape is entered from the input file.

At that time, the ISIS READ routine will read the number of characters specified in the read parameter block into the user's program buffer. The line edit buffer pointer will be posiitoned after the last character read. Further READ calls will read characters beginning at the pointer. New input will not be read into the line edit buffer until all of the original buffer is read.

Line editing is only used for input files, and only with the ISIS READ routine.

To specify to ISIS that a file is to be line edited, the file must be opened for line editing. This is done by using the open system call only changing one parameter. The fourth parameter, the echo file AFTN, must be defined.

An echo file is a file that will receive an echo of the data being entered from the line edited file. The CRT screen is commonly used as the echo file so that the user at the keyboard can see the characters or any changes being entered. (See figure 12.)

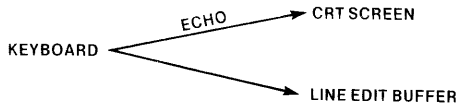

Figure 12

121559-12

If no echo of the input is desired, the byte bucket can be specified as the echo file. The byte bucket (:BB:) is a legal ISIS file that will accept data that is to be discarded. Before using, the byte bucket must be opened with the ISIS OPEN routine.

VIII. LINE EDITING PROGRAM EXAMPLE

The following program demonstrates the use of the console input file, :CI:, as the line edited file. During line editing, input from the console is stored as an intermediate buffer, the line edit buffer. While entering characters into the line edit buffer, it is possible to enter editing characters, making any corrections to the data in the line edit buffer. The editing characters are not stored in the buffer but their function is executed. When a carriage return is entered at the keyboard, or 122 characters have been entered, the input operation from the keyboard is terminated. At that time, the number of characters specified by the READ call are transferred from the line edit buffer to the user's program buffer.

The line edit buffer pointer is now positioned after the last character transferred to the user's program buffer. (See figure 13.)

Subsequent READ calls from the console input file will start transferring characters at the pointer of the line edit buffer. (See figure 14.) New input is not received at the keyboard until the line edit buffer has been cleared. This is done by reading all of the characters from the line edit buffer to the user's program buffer, including the carriage return and line feed.

The example program is invoked from ISIS by entering the name of the located object file and a carriage return on the console keyboard:

-LINEDT <CR>

ISIS uses the :CI: file as a line edited file. When the <CR> is entered after the name of the file, the line edit buffer for :CI: contains LINEDT (see figure 15).

ISIS executes a read characters command until it encounters a space or <CR>. In this example, LINEDT would be sent to the ISIS buffer, and ISIS would load LINEDT and transfer control to the program. The pointer in the line edit buffer would still be positioned at the <CR>. (See figure 16.)

But the LINEDT program will read from the :CI: file, and the CR-LF is still in the line edit buffer of the :CI: file. (See figure 17.)

The user's program that will be reading from :CI:, as LINEDT, must clear the line edit buffer. The line edit pointer is at the <CR> and to clear the buffer, the READ routine must transfer the characters to the user's buffer.

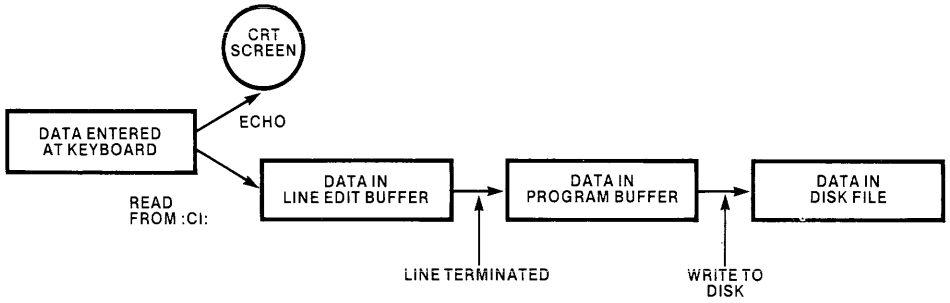

Figure 13

121559-13

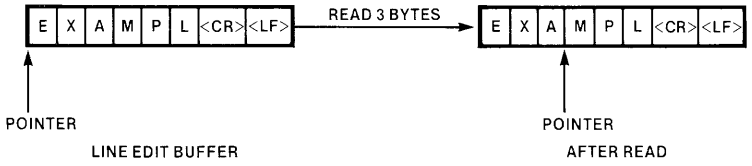

Figure 14

121559-14

Figure 15

121559-15

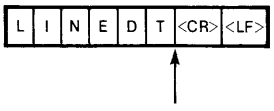

Figure 16

121559-16

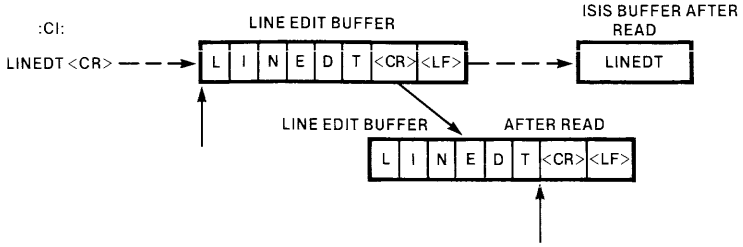

Figure 17

121559-17

In the example program LINEDT, the READ routine is called to clear the buffer. The number of characters specified to READ must be greater than or equal to two. ISIS READ will transfer the <CR><LF> and clear the line edit buffer for further input from :CI:.

NOTE:
The CR-LF is left in the :CI: line edit buffer by ISIS before executing the user's program. Any program that is loaded and executed by ISIS on :CI: and is expecting input from :CI: must read the CR-LF first.

Twenty characters are read from the line edit buffer. It has been cleared of the CR-LF so input will be accepted from the :CI: file.

DATA FROM :CI: -----> LINE EDIT BUFFER

The data from :CI: will be transferred to the line edit buffer until a <CR> is entered at the keyboard, or 122 characters are entered. At this time, the read routine will transfer twenty characters from the line edit buffer to the user's input buffer.

The pointer in the line edit buffer will be at character 21. A call to the WRITE routine is then issued, to write the twenty characters in the user's program buffer to the disk file TST. (See figure 18.)

Another call to the READ routine is made, reading twenty characters from the line edit buffer. (It is assumed that more than twenty characters were entered at the console.) Starting at the character 21, twenty characters will be transferred to the user's program buffer. If there are less than twenty characters remaining, the characters up until the <CR> will be transferred. (See figure 19.)

A second call to the WRITE routine is issued to write the characters in the user's program buffer to the diskette file. The WRITE will begin after the last character of the previous write.

This completes the operation of the example. If further input from the console keyboard is desired, it would be necessary to read all the characters in the line edit buffer, including the <CR> <LF> to clear the buffer. Any further READ calls would input characters from the line edited file.

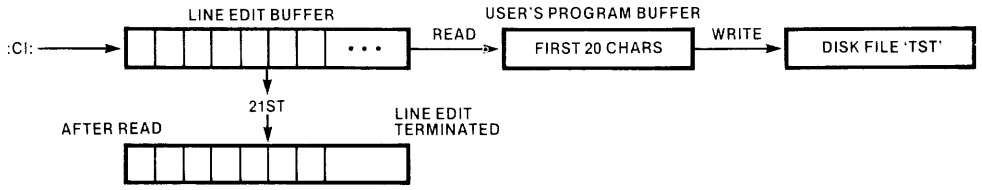

Figure 18

121559-18

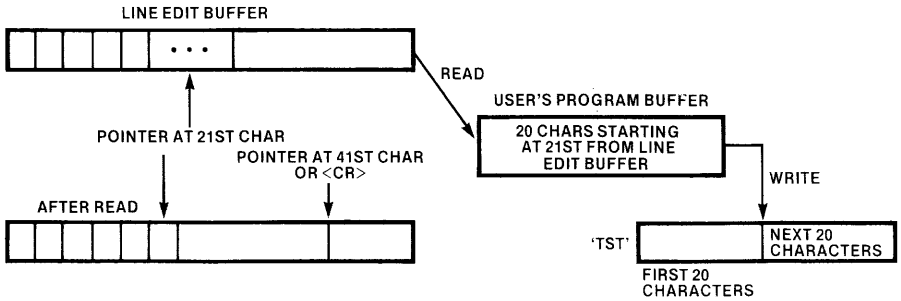

Figure 19

121559-19

```
:F1:ASM80 LINEDT.SRC


ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    1

  LOC  OBJ          LINE        SOURCE STATEMENT

                      1 ;
                      2 ;
                      3 ;  THIS PROGRAM EXEMPLIFIES THE USE OF LINE EDITED DEVICES.  THE
                      4 ;  CONSOLE INPUT DEVICE IS USED AS THE LINE EDITED DEVICE, :CI:.
                      5 ;  THE INPUT IS ECHOED TO THE CONSOLE CRT AND IS WRITTEN TO
                      6 ;  THE DISKETTE FILE 'TST'.
                      7 ;
                      8 ;
                      9 ;
                     10 ;   EXTERNAL DIRECTIVE FOR LINK TO ISIS SYSTEM ROUTINES
                     11 ;
                     12          EXTRN ISIS
                     13 ;
                     14          CSEG
                     15          STKLN   20
                     16 ;
                     17 ;   EQUATE DIRECTIVES FOR SYSTEM ROUTINE IDENTIFIERS
                     18 ;
  0000               19 OPEN     EQU     0
  0003               20 READ     EQU     3
  0004               21 WRITE    EQU     4
  0009               22 EXIT     EQU     9
  000C               23 ERROR    EQU     12
                     24 ;
                     25 ;
                     26 ;   OPEN CALL PARAMETER BLOCK FOR DISK FILE 'TST'
                     27 ;
  0000 0A00      C   28 OBLK:    DW      OAFT
  0002 0C00      C   29          DW      OFILE
  0004 0200          30 ACCESS:  DW      2
  0006 0000          31 ECHO:    DW      0
  0008 9D00      C   32          DW      ESTAT
                     33 ;
                     34 ;  VARIABLES WHOSE ADDRESSES WERE USED IN OPEN PARAMETER BLOCK
                     35 ;
  000A               36 OAFT:    DS      2
  000C 545354        37 OFILE:   DB      'TST'
                     38 ;
                     39 ;  READ ROUTINE PARAMETER BLOCK
                     40 ;
                     41 RBLK:
  000F 0100          42 RAFT:    DW      1
  0011 1B00      C   43          DW      IBUF
  0013 1400          44 RCNT:    DW      20
  0015 1900      C   45          DW      ACTUAL
  0017 9D00      C   46          DW      ESTAT
                     47 ;
                     48 ;   VARIABLES WITH ADDRESSES DEFINED IN THE READ PARAMETER BLOCK
                     49 ;
  0019               50 ACTUAL:  DS      2
  001B               51 IBUF:    DS      122
                     52 ;
                     53 ;
                     54 ;   WRITE ROUTINE PARAMETER BLOCK
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    2

  LOC  OBJ          LINE        SOURCE STATEMENT

                      55 ;
                      56 ;
                      57 WBLK:
  0095                58 WAFT:   DS       2
  0097 1B00      C    59         DW       IBUF
  0099 1400           60 WCNT:   DW       20
  009B 9D00      C    61         DW       ESTAT
                      62 ;
                      63 ;  ERROR ROUTINE PARAMETER BLOCK
                      64 ;
                      65 EBLK:
  009D                66 ESTAT:  DS       2
  009F A100      C    67         DW       STATUS
  00A1                68 STATUS: DS       2
                      69 ;
                      70 ;
                      71 ;  EXIT ROUTINE PARAMETER BLOCK
                      72 ;
  00A3 9D00      C    73 XBLK:   DW       ESTAT
                      74 ;
                      75 ;
                      76 ;   INITIALIZE STACK POINTER
                      77 ;
  00A5 310000    S    78 START:  LXI      SP,STACK
                      79 ;
                      80 ;  OPEN DISKETTE FILE 'TST'
                      81 ;
  00A8 0E00           82         MVI      C,OPEN
  00AA 110000    C    83         LXI      D,OBLK
                      84 ;
                      85 ;  CALL OPEN ROUTINE
                      86 ;
  00AD CD0000    E    87         CALL     ISIS
                      88 ;
                      89 ;  TEST THE STATUS WORD FOR ERROR CONDITION
                      90 ;
  00B0 3A9D00    C    91         LDA      ESTAT
  00B3 B7             92         ORA      A
  00B4 C20B01    C    93         JNZ      EXCEPT
                      94 ;
                      95 ;  THE AFTN RETURNED BY ISIS TO IDENTIFY THE FILE IS MOVED TO THE VARIABLE
                      96 ;  IN THE PARAMETER BLOCK TO IDENTIFY THE FILE FOR THE READ CALL
                      97 ;
                      98 ;
  00B7 2A0A00    C    99         LHLD     OAFT
  00BA 229500    C   100         SHLD     WAFT
                     101 ;
                     102 ;  READ THE CR-LF FROM THE LINE EDIT BUFFER TO CLEAR THE BUFFER
                     103
  00BD 0E03          104         MVI      C,READ
  00BF 110F00    C   105         LXI      D,RBLK
  00C2 CD0000    E   106         CALL     ISIS
                     107 ;
                     108 ;  TEST THE STATUS WORD FOR ERROR CONDITION
                     109 ;
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    3


  LOC  OBJ          LINE        SOURCE STATEMENT

  00C5 3A9D00   C    110          LDA     ESTAT
  00C8 B7            111          ORA     A
  00C9 C20B01   C    112          JNZ     EXCEPT
                     113 ;
                     114 ;  READ FROM LINE EDIT BUFFER FIRST 20 CHARACTERS AND STORE IN IBUF
                     115 ;
  00CC 0E03          116          MVI     C,READ
  00CE 110F00   C    117          LXI     D,RBLK
  00D1 CD0000   E    118          CALL    ISIS
  00D4 3A9D00   C    119          LDA     ESTAT
  00D7 B7            120          ORA     A
  00D8 C20B01   C    121          JNZ     EXCEPT
                     122 ;
                     123 ;
                     124 ;  WRITE IBUF BUFFER TO DISK FILE 'TST'
                     125 ;
  00DB 0E04          126          MVI     C,WRITE
  00DD 119500   C    127          LXI     D,WBLK
  00E0 CD0000   E    128          CALL    ISIS
  00E3 3A9D00   C    129          LDA     ESTAT
  00E6 B7            130          ORA     A
  00E7 C20B01   C    131          JNZ     EXCEPT
                     132 ;
                     133 ;  THE POINTER OF THE LINE EDIT BUFFER IS NOW POSITIONED AT 21ST
                     134 ;  CHARACTER OF THE BUFFER.  THE FOLLOWING READ WILL READ THE
                     135 ;  NEXT 20 CHARACTERS STARTING AT THE 21ST CHARACTER.
                     136 ;
                     137 ;
                     138 ;
  00EA 0E03          139          MVI     C,READ
  00EC 110F00   C    140          LXI     D,RBLK
  00EF CD0000   E    141          CALL    ISIS
                     142 ;
                     143 ;  TEST ERROR STATUS
                     144 ;
  00F2 3A9D00   C    145          LDA     ESTAT
  00F5 B7            146          ORA     A
  00F6 C20B01   C    147          JNZ     EXCEPT
                     148 ;
                     149 ;  WRITE NEXT 20 CHARACTERS TO THE DISK FILE.  THE WRITE WILL BEGIN
                     150 ;  AFTER THE LAST CHARACTERS WRITTEN
                     151 ;
  00F9 0E04          152          MVI     C,WRITE
  00FB 119500   C    153          LXI     D,WBLK
  00FE CD0000   E    154          CALL    ISIS
  0101 3A9D00   C    155          LDA     ESTAT
  0104 B7            156          ORA     A
  0105 C20B01   C    157          JNZ     EXCEPT
  0108 C31301   C    158          JMP     XIT
                     159 ;
                     160 ;  WHEN AN ERROR CONDITION IS ENCOUNTERED, CONTROL IS TRANSFERRED HERE.
                     161 ;
  010B 0E0C          162 EXCEPT: MVI     C,ERROR
  010D 119D00   C    163          LXI     D,EBLK
  0110 CD0000   E    164          CALL    ISIS
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    4


  LOC  OBJ          LINE        SOURCE STATEMENT

                     165 ;
                     166 ;  THE EXIT ROUTINE
                     167 ;
  0113 0E09          168 XIT:   MVI       C,EXIT
  0115 11A300    C   169         LXI       D,XBLK
  0118 CD0000    E   170         CALL      ISIS
                     171 ;
  00A5           C   172         END       START


PUBLIC SYMBOLS


EXTERNAL SYMBOLS
ISIS    E 0000

USER SYMBOLS
ACCESS C 0004    ACTUAL C 0019    EBLK   C 009D    ECHO   C 0006    ERROR  A 000C    FSTAT  C 009D    EXCEPT C 010B
EXIT   A 0009    IBUF   C 001B    ISIS   E 0000    OAFT   C 000A    OBLK   C 0000    OFILE  C 000C    OPEN   A 0000
RAFT   C 000F    RBLK   C 000F    RCNT   C 0013    READ   A 0003    START  C 00A5    STATUS C 00A1    WAFT   C 0095
WBLK   C 0095    WCNT   C 0099    WRITE  A 0004    XBLK   C 00A3    XIT    C 0113

ASSEMBLY COMPLETE,   NO ERRORS
```

U.S. AND CANADIAN SALES OFFICES

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

ALABAMA
Intel Corp.
3322 S. Parkway, Ste. 71
Holiday Office Center
Huntsville 35802
Tel: (205) 883-2430

Pen-Tech Associates, Inc.
Holiday Office Center
3322 Memorial Pkwy., S.W.
Huntsville 35801
Tel: (205) 881-9298

ARIZONA
Intel Corp.
10210 N. 25th Avenue, Suite 11
Phoenix 85021
Tel: (602) 997-9695

BFA
4426 North Saddle Bag Trail
Scottsdale 85251
Tel: (602) 994-5400

CALIFORNIA
Intel Corp.
7670 Opportunity Rd.
Suite 135
San Diego 92111
Tel: (714) 268-3563

Intel Corp.*
1651 East 4th Street
Suite 105
Santa Ana 92701
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
15335 Morrison
Suite 345
Sherman Oaks 91403
(213) 986-9510
TWX: 910-495-2045

Intel Corp.*
3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
TWX: 910-338-0255

Earle Associates, Inc.
4617 Ruffner Street
Suite 202
San Diego 92111
Tel: (714) 278-5441

Mac-I
2576 Shattuck Ave.
Suite 4B
Berkeley 94704
Tel: (415) 843-7625

Mac-I
P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Mac-I
11725 Espen Circle
P.O. Box 8763
Fountain Valley 92708
Tel: (714) 839-3341

Mac-I
110 Sutter Street
Suite 715
San Francisco 94104
Tel: (415) 982-3673

Mac-I
20121 Ventura Blvd., Suite 240E
Woodland Hills 91364
Tel: (213) 347-5900

COLORADO
Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

Westek Data Products, Inc.
25921 Fern Gulch
P.O. Box 1355
Evergreen 80439
Tel: (303) 674-5255

Westek Data Products, Inc.
1322 Arapahoe
Boulder 80302
Tel: (303) 449-2620

Westek Data Products, Inc.
1228 W. Hinsdale Dr.
Littleton 80120
Tel: (303) 797-0482

CONNECTICUT
Intel Corp.
Peacock Alley
1 Padanaram Road, Suite 146
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

FLORIDA
Intel Corp.
1001 N.W. 62nd Street, Suite 406
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

FLORIDA (cont.)
Intel Corp.
5151 Adanson Street, Suite 203
Orlando 32804
Tel: (305) 628-2393
TWX: 810-853-9219

Pen-Tech Associates, Inc.
201 S.E. 15th Terrace, Suite K
Deerfield Beach 33441
Tel: (305) 421-4989

Pen-Tech Associates, Inc.
111 So. Maitland Ave., Suite 202
P.O. Box 1475
Maitland 32751
Tel: (305) 645-3444

GEORGIA
Pen Tech Associates, Inc.
Cherokee Center, Suite 21
627 Cherokee Street
Marietta 30060
Tel: (404) 424-1931

ILLINOIS
Intel Corp.*
2250 Golf Road, Suite 815
Rolling Meadows 60008
Tel: (312) 981-7200
TWX: 910-651-5881

First Rep Company
9400-9420 W. Foster Avenue
Chicago 60656
Tel: (312) 992-0830
TWX: 910-227-4927

Technical Representatives
1502 North Lee Street
Bloomington 61701
Tel: (309) 829-8080

INDIANA
Intel Corp.
2212 Maplecrest Rd.
Ft. Wayne 46815
Tel: (219) 493-2571

IOWA
Technical Representatives, Inc.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52405
Tel: (319) 393-5510

KANSAS
Intel Corp.
9393 W. 110th St., Ste. 265
Overland Park 66210
Tel: (913) 642-8080

Technical Representatives, Inc.
8245 Nieman Road, Suite #100
Lenexa 66214
Tel: (913) 888-0212, 3, & 4
TWX: 910-749-6412

KENTUCKY
Lowry & Associates Inc.
P.O. Box 1827
Lexington 40593
Tel: (606) 273-3771

MARYLAND
Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Glen White Associates
57 W. Timonium Road, Suite 307
Timonium 21093
Tel: (301) 252-6360

Mesa Inc.
11900 Parklawn Drive
Rockville 20852
Tel: Wash. (301) 881-8430
Balto. (301) 792-0021

MASSACHUSETTS
Intel Corp.*
27 Industrial Ave.
Chelmsford 01824
Tel: (617) 667-8126
TWX: 710-343-6333

EMC Corp.
381 Elliot Street
Newton 02164
Tel: (617) 244-4740

MICHIGAN
Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 910-420-1212
TELEX: 2 31143

Lowry & Associates, Inc.
135 W. North Street
Suite 4
Brighton 48116
Tel: (313) 227-7067

Lowry & Associates, Inc.
3902 Costa NE
Grand Rapids 49505
Tel: (616) 363-9839

MINNESOTA
Intel Corp.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

Dytek North
1821 University Ave.
Room 163N
St. Paul 55104
Tel: (612) 645-5816

MISSOURI
Technical Representatives, Inc.
320 Brookes Drive, Suite 104
Hazelwood 63042
Tel: (314) 731-5200
TWX: 910-762-0618

NEW JERSEY
Intel Corp.*
Raritan Plaza
2nd Floor
Raritan Center
Edison 08817
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO
BFA Corporation
P.O. Box 1237
Las Cruces 88001
Tel: (505) 523-0601
TWX: 910-983-0543

BFA Corporation
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK
Intel Corp.*
350 Vanderbilt Motor Pkwy.
Suite 402
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington St.
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
2255 Lyell Avenue
Lower Floor East Suite
Rochester 14606
Tel: (716) 254-6120
TWX: 510-253-7391

Measurement Technology, Inc.
159 Northern Boulevard
Great Neck 11021
Tel: (516) 482-3500

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
2 E. Main
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA
Intel Corp.
154 Huffman Mill Rd.

Tel: (919) 584-3631

Pen-Tech Associates, Inc.
1202 Eastchester Dr.
Highpoint 27260
Tel: (919) 883-9125

Glen White Associates
4009 Barrett Dr.
Raleigh 27609
Tel: (919) 787-7016

OHIO
Intel Corp.*
6500 Poe Avenue
Dayton 45415
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg. #210
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736

Lowry & Associates Inc.
1440 Snow Road
Suite 216
Cleveland 44134
Tel: (216) 398-0506

Lowry & Associates, Inc.
1524 Marsetta Drive
Dayton 45432
Tel: (513) 429-9040

Lowry & Associates, Inc.
2735 Cleveland Ave.
Columbus 43224
Tel: (614) 436-2051

OREGON
Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
Beaverton 97005
Tel: (503) 641-8086

PENNSYLVANIA
Intel Corp.*
275 Commerce Dr.
200 Office Center
Suite 300
Fort Washington 19034
Tel: (215) 542-9444
TWX: 510-661-2077

Lowry & Associates, Inc.
Seven Parkway Center
Suite 455
Pittsburgh 15520
Tel: (412) 922-5110

Q.E.D. Electronics
300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS
Intel Corp.*
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-9521
TWX: 910-860-5487

Intel Corp.*
6420 Richmond Ave.
Suite 280
Houston 77057
Tel: (713) 784-3400

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 988-9421

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

VIRGINIA
Glen White Associates
Route 2, Box 193
Charlottesville 22901
Tel: (804) 295-7686

Glen White Associates
P.O. Box 10186
Lynchburg 24506
Tel: (804) 384-6920

Glen White Associates
Rt. #1, Box 322
Colonial Beach 22442
Tel: (804) 224-7764

WASHINGTON
Intel Corp.
Suite 114 Bldg. 3
1603 116th Ave. N.E.
Bellevue 98005
Tel: (206) 453-8086

WISCONSIN
Intel Corp.
150 S. Sunnyslope Rd.
Brookfield 53005
Tel: (414) 784-9060

First Rep Company
9401 W. Beloit Rd.
Suite 304
Milwaukee 53227
Tel: (414) 546-2033

CANADA
Intel Semiconductor Corp.*
Suite 233, Bell Mews
39 Highway 7, Bells Corners
Ottawa, Ontario K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor Corp.
50 Galaxy Blvd.
Unit 12
Rexdale, Ontario
M9W 4Y5
Tel:(416) 675-2105
TELEX: 06983574

Multilek, Inc.*
15 Grenfell Crescent
Ottawa, Ontario K2G 0G3
Tel: (613) 226-2365
TELEX: 053-4585

Multilek, Inc.
Toronto
Tel: (416) 245-4622

Multilek, Inc.
Montreal
Tel: (514) 481-1350

*Field application location

intel®

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

ALABAMA

†Hamilton/Avnet Electronics
4692 Commercial Drive N.W.
Huntsville 35805
Tel: (205) 837-7210

†Pioneer
1207 Putman Drive NW
Huntsville 35805
Tel: (205) 837-9033
TWX: 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
2615 South 21st Street
Phoenix 85034
Tel: (602) 275-7851

†Wyle Distribution Group
8155 N. 24th Avenue
Phoenix 85021
Tel: (602) 249-2232
TWX: 910-951-4282

CALIFORNIA

†Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel: (714) 754-6111
TWX: 910-595-1928

Hamilton/Avnet
1175 Bordeaux Dr.
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332

†Hamilton/Avnet Electronics
8917 Complex Drive
San Diego 92123
Tel: (714) 279-2421
TWX: 910-335-1216

Hamilton/Avnet
10912 W. Washington Blvd.
Culver City 90230
Tel: (213) 558-2809 (2665)
TWX: 910-340-6364 or 7073

†Hamilton Electro Sales
10912 W. Washington Boulevard
Culver City 90230
Tel: (213) 558-2121

†Wyle Distribution Group
124 Maryland Street
El Segundo 90245
Tel: (213) 322-3826
TWX: 910-348-7140 or 7111

†Wyle Distribution Group
9525 Chesapeake Dr.
San Diego 92123
Tel: (714) 565-9171
TWX: 910-335-1590

†Wyle/Elmar
3000 Bowers Avenue
Santa Clara 95052
Tel: (408) 727-2500
TWX: 910-338-0451 or 0296

Hamilton/Avnet Electronics
17312 Eastman Street
Irvine 92714
Tel: (714) 979-6864

COLORADO

†Wyle/Elmar/Denver
6777 E. 50th Avenue
Commerce City 80022
Tel: (303) 287-9611
TWX: 910-931-0510

†Hamilton/Avnet Electronics
5921 No. Broadway
Denver 80216
Tel: (303) 534-1212
TWX: 910-931-0510

CONNECTICUT

†Arrow/Connecticut
P.O. Box 5003
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
TWX: 710-476-0162

†Hamilton/Avnet Electronics
643 Danbury Road
Georgetown 06829
Tel: (203) 762-0361

†Harvey Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
TWX: 710-468-3373

FLORIDA

Arrow Electronics
1001 N.W. 62nd Street
Suite 108
Ft. Lauderdale 33309
Tel: (305) 776-7790

Arrow Electronics
115 Palm Bay Road, NW
Suite 10, Bldg. 200
Palm Bay 32905
Tel: (305) 725-1480
TWX: 510-959-6337

†Hamilton/Avnet Electronics
6800 Northwest 20th Ave.
Ft. Lauderdale 33309
Tel: (305) 971-2900
TWX: 510-955-3097

†Pioneer
6220 S. Orange Blossom Trail
Suite 412
Orlando 32809
Tel: (305) 859-3600
TWX: 810-850-0177

Hamilton/Avnet
3197 Tech. Drive N.
St. Petersburg 33702
Tel: (813) 576-3930
TWX: 810-863-0374

GEORGIA

Arrow Electronics
2979 Pacific Drive
Norcross 30071
Tel: (404) 449-8252
TWX: 810-757-4213

†Hamilton/Avnet Electronics
6700 I-85 Access Road, #11
Suite IE
Norcross 30071
Tel: (404) 448-0800

ILLINOIS

Arrow Electronics
492 Lunt Avenue
P.O. Box 94248
Schaumburg 60193
Tel: (312) 893-9420
TWX: 910-222-1807

†Hamilton/Avnet Electronics
3901 No. 25th Ave.
Schiller Park 60176
Tel: (312) 678-6310
TWX: 910-227-0060

Pioneer/Chicago
1511 Carmen Drive
Elk Grove 60006
Tel: (312) 437-9680
TWX: 910-222-1834

INDIANA

†Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300
TWX: 810-260-1794

KANSAS

†Hamilton/Avnet Electronics
9219 Quivira Road
Overland Park 66215
Tel: (913) 888-8900

MARYLAND

†Hamilton Avnet
P.O. Box 647,
BWI Airport
7235 Standard Drive
Hanover 21076
Tel: (301) 796-5684
TWX: 710-862-1861

†Pioneer/Washington
9100 Gaither Road
Gaithersburg 20760
Tel: (301) 948-0710
TWX: 710-828-0545

MASSACHUSETTS

†Hamilton/Avnet Electronics
50 Tower Office Park
Woburn 01801
Tel: (617) 935-9700

MASSACHUSETTS (continued)

Harvey/Boston
44 Hartwell Ave.
Lexington 02173
Tel: (617) 861-9200

MICHIGAN

†Arrow Electronics
3921 Varsity Road
Ann Arbor 48140
Tel: (313) 971-8220
TWX: 810-223-6020

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
TWX: 810-242-3271

†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-242-8775

MINNESOTA

†Industrial Components
5229 Edina Industrial Blvd.
Minneapolis 55435
Tel: (612) 831-2666
TWX: 910-576-3153

Arrow Electronics
5251 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-576-2726

†Hamilton/Avnet Electronics
7449 Cahill Road
Edina 55435
Tel: (612) 941-3801
TWX: 910-576-2720

MISSOURI

†Hamilton/Avnet Electronics
396 Brookes Drive
Hazelwood 63042
Tel: (314) 731-1144
TWX: 910-762-0606

NEW JERSEY

Arrow/Philadelphia
Pleasant Valley
Moorestown 08057
Tel: (215) 928-1800
TWX: 710-897-0829

Arrow Electronics
285 Midland Avenue
Saddlebrook 07662
Tel: (201) 797-5800
TWX: 710-988-2206

Hamilton/Avnet
10 Industrial
Fairfield 07006
Tel: (201) 575-3390
TWX: 710-734-4338

†Harvey Electronics
45 Route 46
Pinebrook 07058
Tel: (201) 227-1262
TWX: 710-734-4382

†Hamilton/Avnet Electronics
1 Keystone Ave.
Cherry Hill 08003
Tel: (609) 424-0100
TWX: 710-897-1405

NEW MEXICO

†Alliance Electronics Inc.
11721 Central Ave.
Albuquerque 87123
Tel: (505) 292-3360
TWX: 910-989-1151

†Hamilton/Avnet Electronics
2524 Baylor Drive, S.E.
Albuquerque 87119
Tel: (505) 765-1500

†Microcomputer System Technical Demonstrator Centers

intel®

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

NEW YORK

Harvey Electronics
P.O. Box 1208
Binghampton 13902
Tel: (607) 748-8211
TWX: 510-252-0893

Arrow Electronics
900 Broad Hollow Road
Farmingdale 11735
Tel: (516) 694-6800
TWX: 510-224-6494

†Arrow/Rochester
3000 South Winton Road
Rochester 14623
Tel: (716) 275-0300
TWX: 910-338-0026

†Hamilton/Avnet Electronics
167 Clay Road
Rochester 14623
Tel: (716) 475-9130
TWX: 910-340-6364

†Arrow/Syracuse
7705 Maltlage Drive
Liverpool 13088
Tel: (315) 652-1000
TWX: 710-545-0230

Arrow Electronics
20 Oser Avenue
Hauppauge 11787
Tel: (516) 231-1000
TWX: 510-224-6494

†Hamilton/Avnet Electronics
16 Corporate Circle
E. Syracuse 13057
Tel: (315) 437-2641

†Hamilton/Avnet Electronics
70 State Street
Westbury, L.I. 11590
Tel: (516) 454-6000
TWX: 510-252-0893

†Harvey Electronics
60 Crossways Park West
Woodbury 11797
Tel: (516) 921-8700
TWX: 510-221-2184

Harvey/Rochester
10 Riverton Way
West Henrietta 14586
Tel: (716) 334-5920

NORTH CAROLINA

Pioneer/Carolina
106 Industrial Ave.
Greensboro 27406
Tel: (919) 273-4441
TWX: 510-925-1114

†Hamilton/Avnet Electronics
2803 Industrial Drive
Raleigh 27609
Tel: (919) 829-8030

Arrow Electronics
938 Burke Street
Winston-Salem 27102
Tel: (919) 996-2039
TWX: 510-922-4765

OHIO

Arrow Electronics
3100 Plainfield Road
Kettering 45432
Tel: (513) 253-9176
TWX: 810-459-1611

Arrow Electronics
6238 Cochran Rd.
Solon 44139
Tel: (216) 248-3990

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel: (513) 433-0610
TWX: 910-340-2531

†Pioneer/Dayton
1900 Troy Street
Dayton 45404
Tel: (513) 236-9900
TWX: 810-459-1622

Arrow Electronics
10 Knollcrest Dr.
Reading 44139
Tel: (513) 761-5432
TWX: 810-461-2670

OHIO (continued)

†Pioneer/Cleveland
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
TWX: 810-422-2210

†Hamilton/Avnet Electronics
4588 Emory Industrial Parkway
Warrensville Heights 44128
Tel: (216) 831-3500

OKLAHOMA

†Components Specialties, Inc.
7920 E. 40th Street
Tulsa 74145
Tel: (918) 664-2820
TWX: 910-845-2215

OREGON

†Almac/Stroum Electronics
8022 S.W. Nimbos, Bldg. 7
Beaverton 97005
Tel: (503) 641-9070

PENNSYLVANIA

Pioneer/Pittsburgh
560 Alpha Drive
Pittsburgh 15238
Tel: (412) 782-2300
TWX: 710-795-3122

Pioneer/Delaware Valley
141 Gibraltar Road
Horsham 19044
Tel: (215) 674-4000
TWX: 510-665-6778

TENNESSEE

Arrow/Knoxville
P.O. Box 129
W. Andrew Johnson Hwy.
Talbott 37677
Tel: (615) 587-2137

TEXAS

Component Specialties Inc.
8222 Jamestown Drive
Suite 115
Austin 78758
Tel: (512) 837-8922
TWX: 910-874-1320

Hamilton/Avnet
10508A Boyer Blvd.
Austin 78757
Tel: (512) 837-8911

†Hamilton/Avnet Electronics
4445 Sigma Road
Dallas 75240
Tel: (214) 661-8661
TWX: 910-860-5371

†Hamilton/Avnet Electronics
3939 Ann Arbor
Houston 77063
Tel: (713) 780-1771

†Component Specialties, Inc.
10807 Shady Trail, Suite 101
Dallas 75220
Tel: (214) 357-6511
TWX: 910-861-4999

†Component Specialties, Inc.
8585 Commerce Park Drive, Suite 590
Houston 77036
Tel: (713) 771-7237
TWX: 910-881-2422

Arrow Electronics
13715 Gamma Road
Dallas 75234
Tel: (214) 386-7500
TWX: 910-861-5495

UTAH

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City, 84119
Tel: (801) 972-2800

WASHINGTON

†Hamilton/Avnet Electronics
14212 N.E. 21st
Bellevue 98005
Tel: (206) 746-8750

†Almac/Stroum Electronics
5811 Sixth Ave. South
Seattle 98108
Tel: (206) 763-2300
TWX: 910-444-2067

WASHINGTON (continued)

†Liberty Electronics
1750 132nd Avenue NE
Bellevue 98005
Tel: (206) 453-8300
TWX: 910-443-2526

WISCONSIN

Arrow Electronics
434 W. Rausson Avenue
Oak Creek 53154
Tel: (414) 764-6600
TWX: 910-338-0026

†Hamilton/Avnet
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510
TWX: 910-262-1182

CANADA

ALBERTA

Zentronics
9224 27th Avenue
Edmonton T6N 1B2

†L. A. Varah Ltd.
4742 14th Street N.E.
Calgary T2E 6L7
Tel: (403) 230-1235
TWX: 018-258-97

Zentronics
3651 21st N.E.
Calgary T2E 6T5
Tel: (403) 230-1422

BRITISH COLUMBIA

†L.A. Varah Ltd.
2077 Alberta Street
Vancouver V5Y 1C4
Tel: (604) 873-3211
TWX: 610-929-1068

Zentronics
550 Cambie St.
Vancouver V6B 2N7
Tel: (604) 688-2533
TWX: 04-5077-89

MANITOBA

L. A. Varah
1-1832 King Edward Street
Winnipeg R2R 0N1
Tel: (204) 633-6190
TWX: 07-55-365

Zentronics
590 Berry St.
Winnipeg R3H 0S1

ONTARIO

†L.A. Varah, Ltd.
505 Kenora Avenue
Hamilton L8E-3P2
Tel: (416) 561-9311
TWX: 061-8349

†Hamilton/Avnet Electronics
3688 Nashua Drive, Units G & H
Mississauga L4V IM5
Tel: (416) 677-7432
TWX: 610-492-8860

†Hamilton/Avnet Electronics
1735 Courtwood Cresc.
Ottawa K2C 3J2
Tel: (613) 226-1700

‡Zentronics
141 Catherine Street
Ottawa, Ontario K2P 1C3
Tel: (613) 238-6411
TWX: 053-3636

‡Zentronics
1355 Meyerside Drive
Mississauga, Ontario L5T IC9
Tel: (416) 676-9000
Telex: 06-983-657

QUEBEC

†Hamilton/Avnet Electronics
2670 Paulus Street
St. Laurent H4S 1G2
Tel: (514) 331-6443
TWX: 610-421-3731

Zentronics
5010 Pare Street
Montreal H4P 1P3
Tel: (514) 735-5361
TWX: 05-827-535

†Microcomputer System Technical Demonstrator Centers

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 987-8080

Printed in U.S.A./H230/0180/5K/TP/NCG