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

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

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and its affiliates and may be used to identify Intel products:

- BXEP
- CREDI
- i
- FICE
- ICE
- iCS
- IMX
- IMX
- Intel
- Intel
- ISBC
- iSBX
- iS
- iS
- ISM
- iSP
- iSP
- iSP
- iSP
- IMX
- IMX
- IMX
- IMX
- MULTICHANNEL
- MULTIMODULE
- Plug-A-Bubble
- PROMPT
- PROMPT
- PROMPT
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
- RMX
<table>
<thead>
<tr>
<th>REV.</th>
<th>REVISION HISTORY</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-001</td>
<td>Original issue.</td>
<td>8/82</td>
</tr>
</tbody>
</table>
This manual describes the CRT and keyboard interface for the enhanced Series II
development system.

It is divided into four chapters and two appendixes.

Chapter 1, “Introduction,” briefly describes the capabilities of the enhanced Series II
CRT and keyboard interface.

Chapter 2, “Preprogrammed Keys,” lists the preprogrammed keys and their associated
keywords.

Chapter 3, “Console Output Functions,” describes the console output function codes
with examples.

Chapter 4, “Block Movement of Data to the CRT,” describes the IOC commands
(with examples) that allow block movement of data to the CRT.

Appendix A, “Keyboard Codes Generated,” lists the hexadecimal values of the Intel
keyboard characters.

Appendix B, “ASCII Character Set,” lists the ASCII codes with their hexadecimal
values.

Related Publications

*Model 511 IOC Firmware Enhancement Kit Installation Instructions*, order number
122014

*Intec Dynamic Microcomputer Development System Hardware Interface Manual*,
order number 9800555

order number 9800556

*Component Data Catalog*, order number 210298

Notational Conventions

**UPPERCASE** Indicates that characters shown in uppercase must be entered
in the order shown.

*italic* Indicates variable information.

[ ] Indicate optional arguments or parameters.
The Series II CRT and keyboard are controlled by an Input/Output Controller Board, which is referred to as the IOC throughout this manual. All models of the Series II development systems provide cursor movement, clear screen and clear line functions, and display of the ASCII character set.

But the enhanced Series II provides much more:
- Automatic keystroke repeat function
- Preprogrammed keystroke sequences
- New console output functions, including direct cursor addressing
- Block movement of data to the CRT

These functions are provided by new IOC and keyboard firmware. (The cursor now appears as a solid, non-blinking block.)

If you have purchased the new firmware as an update package (iMDX 511), you will receive:
- Two single-density diskettes
- Two double-density diskettes
- Four 2716 PROMs (IOC CRT firmware)
- One 8741A PROM (IOC keyboard firmware)
- Two key caps (labeled FUNC)

Firmware installation is described in the *Model 511 IOC Firmware Enhancement Kit Installation Instructions*, order number 122014.

If you have purchased a new development system, the PROMS and key cap have already been integrated into your system.
The enhanced Series II IOC firmware provides numerous preprogrammed keys. When these keys and the special function key, FUNC, are pressed simultaneously, preprogrammed character sequences are transmitted as keyboard input. (The preprogrammed keys are listed below.)

For example, if you type FUNC A, the keyboard will respond as if you typed ALTER followed by a blank. Upper and lowercase are significant when typing preprogrammed keys. That is, if you type FUNC a, the keyboard will transmit alter in lowercase letters.

The HELP command, executed by typing FUNC H, displays all of the preprogrammed keys and their associated keywords on screen. When you invoke the HELP menu, the screen is cleared of current text. When you type any character, the HELP menu disappears and previous text is restored to the screen.

**Keys**

Quote marks are used to show significant spaces; they are not transmitted. (Remember, you must press FUNC simultaneously with the key you need.)

```
A = "ALTER 
C = "COPY 
D = "DIR 
E = "CREDIT 
I = "ATTRIB 
J = "JOB 
K = "DELETE 
L = ";LP: " 
M = "LOGON 
N = "ASSIGN 
O = "LOGOFF 
P = ";SP: " 
R = "RUN 
S = "SUBMIT 
T = "; TO " 
U = "ACCESS 
X = "EXPORT 
```

\[ n = :Fn: \text{(where } n = (0,1,\ldots,9)\text{)} \]

\[ \text{SHIFT} + n = /JOBn <\text{cr}> \text{(where } n = (0,1,\ldots,9)\text{)} \]

\[ \text{H} = \text{HELP} \rightarrow \text{displays all of the function keys and their preprogrammed keywords} \]
Keywords

ACCESS = U
ALTER = A
ASSIGN = N
ATTRIB = I
COPY = C
CREDIT = E
DELETE = K
DIR = D
EXPORT = X
:Fn. = n (where n = (0,1,...,9))
HELP = H
JOB = J
/JOBn = SHIFT + n (where n = (0,1,...,9))
LOGON = M
LOGOFF = O
:LP: = L
RUN = R
:SP: = P
SUBMIT = S
TO = T

Repeat Function

To send multiple characters, simply hold down the key that you wish to repeat and it will repeat automatically. (The key will repeat after it is held down for 0.5 seconds and will continue to repeat at a rate of 15 characters per second until it is released.)
New Console Output Codes

The IOC firmware modifications provide several new console output functions.

The console output functions are obtained by sending the following codes to console output.

**ESC R**

Clear text from current cursor position to the end of the line.

**ESC S**

Clear text from current cursor position to the end of the screen.

**ESC T**

Clear entire screen (cursor remains in current position).

**ESC Y**

Allows direct cursor addressing.

This command requires coordinates for the row number and column number. (Row number is the row address + 20H; column number is the column address + 20H.)

The command has the following form:

\[ \text{ESC Y} \, r \, c \]

where

\[ r \text{ and } c \]

are variables for row number and column number. If a value given for \( r \) or \( c \) is too large (greater than 38H for row; greater than 6FH for column), the current coordinate is used.

**Example 1**

\[ 1BH \, 59H \, 20H \, 20H \]

This command moves the cursor to the upper left corner of the screen. (20H, 20H indicates the upper left corner of the screen.)
ESC W

ESC W is a generalized insert and delete line function.

The command has the following form:

```
ESC W i d
```

where

```
i and d are line numbers taken from table 3-1.
```

The command inserts a blank line at line i on the screen and deletes the line at line d. All intervening lines scroll in the direction of the deleted line. The cursor moves to the start of the current line.

Line numbers can be presented in three ways: relative to the top of the screen, relative to the bottom of the screen, and relative to the current cursor position. Given that the CRT has exactly 25 lines, the relative-to-bottom settings are redundant. They are provided so that your cursor control sequence will be compatible with possible future terminals with other than 25 lines.

**Example 1**

```
1BH 57H 60H 3FH
```

This command inserts a blank line at the current cursor position.

**Example 2**

```
1BH 57H 3FH 60H
```

This command deletes the line at the current cursor position.

Table 3-1 lists the possible values for i and d.

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>Top line</td>
</tr>
<tr>
<td>01H</td>
<td>One below top</td>
</tr>
<tr>
<td>02H</td>
<td>Second below top</td>
</tr>
<tr>
<td>. .</td>
<td>.</td>
</tr>
<tr>
<td>17H</td>
<td>23rd from top</td>
</tr>
<tr>
<td>18H</td>
<td>24th from top</td>
</tr>
<tr>
<td>27H</td>
<td>24th from bottom</td>
</tr>
<tr>
<td>. .</td>
<td>.</td>
</tr>
<tr>
<td>3EH</td>
<td>One from bottom</td>
</tr>
<tr>
<td>3FH</td>
<td>Bottom line</td>
</tr>
<tr>
<td>47H</td>
<td>24th line above current cursor position</td>
</tr>
<tr>
<td>. .</td>
<td>.</td>
</tr>
<tr>
<td>5FH</td>
<td>Line above current cursor position</td>
</tr>
<tr>
<td>60H</td>
<td>Line at current cursor position</td>
</tr>
<tr>
<td>61H</td>
<td>Line below current cursor position</td>
</tr>
<tr>
<td>. .</td>
<td>.</td>
</tr>
<tr>
<td>78H</td>
<td>24th line below current cursor position</td>
</tr>
</tbody>
</table>
ESCAPE X

Allows you to set the following flags: USES$BITSS$FLAG and PAUSE$FLAG.

The command has the following form:

\texttt{ESC X n s}

where

\begin{align*}
\text{n} & \quad \text{is the flag number.} \\
\text{s} & \quad \text{is the setting (1 = true; 0 = false).}
\end{align*}

USES$BITSS$FLAG

Eight-bit codes are used to take advantage of the field attribute features provided by the 8275 CRT controller chip. (See the Component Data Catalog for information on the 8275 chip.)

The values in table 3-2, when sent to the CRT, are called “attribute bytes.” Wherever they are placed on the CRT screen, a blank will appear in that position. Each succeeding position, until the next attribute byte appears, will have the attribute given.

Attributes can be combined by ORing together the appropriate attribute bytes; e.g., 92H causes blinking characters and reverse video. If blinking and reverse video are enabled simultaneously, only the characters will blink.

USES$BITSS$FLAG is set false to accommodate programs that send the eighth bit as a parity bit.

The following two ESC sequences are used to set USES$BITSS$FLAG:

1. \texttt{1BH 58H 00H 01H}

   All 8 bits of bytes received for console output are used.

2. \texttt{1BH 58H 00H 00H}

   The top bit is masked away (default setting).

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Attribute Byte} & \textbf{Attribute} \\
\hline
80H & None \\
82H & Blinking characters \\
90H & Reverse video \\
A0H & Underline \\
\hline
\end{tabular}
\end{table}

PAUSE$FLAG

The following two ESC sequences are used to set PAUSE$FLAG:

1. \texttt{1BH 58H 01H 01H}
Enter PAUSE mode.
- The IOC pauses (approximately five seconds) every time it is about to scroll the screen (when 20 lines have been output without any keyboard input).
- CNTL S stops output until something other than CNTL S is typed.
- If CNTL Q is typed, or if five seconds elapse, the screen will continue scrolling.

2. 1BH 58H 01H 00H

No PAUSE mode (default setting).

Existing Console Output Codes

This chapter documents the console output functions available with the previous IOC firmware as well as the new firmware. Codes with the same function are grouped together; i.e., ESC A and up arrow are different codes, but they perform the same function.

ESC A
up arrow

Move cursor up. If the cursor is at the top of the screen, the screen wraps and the cursor appears at the bottom of the screen.

ESC B
down arrow
CNTL J

Move cursor down. If the cursor is at the bottom of the screen, the screen scrolls.

ESC C
right arrow

Move cursor right. If the cursor is at the end of the line, it moves to the beginning of the next line. If the cursor is at the end of the screen, the screen scrolls.

ESC D
left arrow

Move cursor left. If the cursor is at the start of the line, the cursor moves to the end of the previous line. If the cursor is at the start of the screen, the cursor moves to the end of the screen.

ESC E

Clear entire screen; cursor moves to the beginning of the screen.

ESC H
HOME

Move the cursor to the top left corner of the screen.
ESC J
Clear text from the beginning of the current line (the line on which the cursor is positioned) to the end of the screen.

ESC K
Clear entire line.

CNTL G
Ring bell.

RETURN
Move the cursor to the start of the following line.

RUBOUT
Do nothing.

All other control keys are ignored.

All other keys display the ASCII character.

Table 3-3 lists the standard CRT output controls, their hexadecimal values, and functions.
### Table 3-3. Standard CRT Output Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Hexadecimal Value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC A</td>
<td>1B 41</td>
<td>Move cursor up.</td>
</tr>
<tr>
<td>up arrow</td>
<td>1E</td>
<td>Move cursor up.</td>
</tr>
<tr>
<td>ESC B</td>
<td>1B 42</td>
<td>Move cursor down.</td>
</tr>
<tr>
<td>down arrow</td>
<td>1C</td>
<td>Move cursor down.</td>
</tr>
<tr>
<td>CNTL J</td>
<td>0A</td>
<td>Move cursor down.</td>
</tr>
<tr>
<td>ESC C</td>
<td>1B 43</td>
<td>Move cursor right.</td>
</tr>
<tr>
<td>right arrow</td>
<td>14</td>
<td>Move cursor right.</td>
</tr>
<tr>
<td>ESC D</td>
<td>1B 44</td>
<td>Move cursor left.</td>
</tr>
<tr>
<td>left arrow</td>
<td>1F</td>
<td>Move cursor left.</td>
</tr>
<tr>
<td>ESC H</td>
<td>1B 48</td>
<td>Move cursor to beginning of screen.</td>
</tr>
<tr>
<td>HOME</td>
<td>1D</td>
<td>Move cursor to beginning of screen.</td>
</tr>
<tr>
<td>RETURN</td>
<td>0D</td>
<td>Move cursor to start of line.</td>
</tr>
<tr>
<td>ESC Y *</td>
<td>1B 59 r c</td>
<td>Move cursor to row r, column c.</td>
</tr>
<tr>
<td>ESC K</td>
<td>1B 4B</td>
<td>Clear line.</td>
</tr>
<tr>
<td>ESC R *</td>
<td>1B 52</td>
<td>Clear text from current cursor position to end of line.</td>
</tr>
<tr>
<td>ESC J</td>
<td>1B 4A</td>
<td>Clear text from current line to end of screen.</td>
</tr>
<tr>
<td>ESC S *</td>
<td>1B 53</td>
<td>Clear text from current cursor position to end of screen.</td>
</tr>
<tr>
<td>ESC T *</td>
<td>1B 54</td>
<td>Clear screen; cursor remains in current cursor position.</td>
</tr>
<tr>
<td>ESC E</td>
<td>1B 45</td>
<td>Clear screen; cursor moves to beginning of screen.</td>
</tr>
<tr>
<td>ESC W *</td>
<td>1B 57 i d</td>
<td>Insert a line at line i; delete a line at line d.</td>
</tr>
<tr>
<td>ESC X *</td>
<td>1B 58 n s</td>
<td>Set flags: USE$BITS$FLAG and PAUSE$FLAG; n = flag number; s = setting.</td>
</tr>
<tr>
<td>CNTL G</td>
<td>07</td>
<td>Ring bell.</td>
</tr>
<tr>
<td>RUBOUT</td>
<td>7F</td>
<td></td>
</tr>
</tbody>
</table>

* Indicates new console output codes.
To allow fast block movement of data to the CRT, several new IOC commands have been implemented as the Block Write function.

The Block Write function is invoked from application programs running on an enhanced Series II. It cannot be invoked through the keyboard.

The Block Write function is initiated by sending a 0FH, 2FH, 4FH, or 6FH command to the command port of the IOC.

The commands 0FH or 4FH require the line number and column number of the position on the screen where the data will be output to the data port following a 0FH or 4FH command. (0,0 is the top left corner of the screen.)

The commands 2FH or 6FH output data at the current cursor position. Table 4-1 summarizes the differences between the commands.

Data is transmitted as follows:

1. The Block Write command (0FH, 2FH, 4FH, or 6FH) is given to the IOC command port.

2. If the command is 0FH or 4FH, the row and column number are output to the IOC data port (00H 00H = upper left corner). If the values for row number and column number are too large (greater than 18H for row number; greater than 4FH for column number), the current coordinates are used.

3. A stream of console output data bytes is output to the IOC data port. If bytes are deposited to the end of the screen, the cursor wraps around to the top of the screen. Bytes are interpreted as follows:

   • Bytes less than 80H are deposited directly in screen memory, except for carriage return and line feed, which have their usual functions.

   • Bytes between 80H and 0FH are interpreted as follows:
     If the command is 0FH or 2FH, byte n is treated as (n−80H) blanks.
     If the command is 4FH or 6FH, the byte is an attribute byte to be deposited directly in screen memory. (See Chapter 3, section “USE$8$BIT$S$FLAG” for a description of attribute bytes.)

   • Byte 0FEH is dropped, but the next byte is placed in screen memory no matter what it is.

   • Byte 0FFH terminates the data and the command unless literalized as described above.

Figure 4-1 is a sample assembly language program that illustrates how to send commands and data to the IOC.

This program uses the following interface procedures to access the IOC commands:

• PCIOC transfers command byte to the IOC.
• PDIOC transfers each following data byte to the IOC.
• GDIOC retrieves data bytes from the IOC.
If there is a possibility of an interrupt procedure (or another processor) accessing the IOC, the interrupts should be disabled with a DI instruction at the beginning of the procedure and enabled with an EI instruction at the end of the procedure. Or, the interrupts can be masked at the beginning of the procedure and unmasked at the end.

Interrupts in this category include pressing Interrupt 0 on the Series II console, since the Series II monitor performs console output.

Similarly, the Series III RUN program communicates with the IOC and therefore precludes 8086 application programs from directly accessing the IOC.

<table>
<thead>
<tr>
<th>Command</th>
<th>Requires Coordinates</th>
<th>Action for High Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F</td>
<td>Yes</td>
<td>(n–80H) blanks</td>
</tr>
<tr>
<td>2F</td>
<td>No</td>
<td>(n–80H) blanks</td>
</tr>
<tr>
<td>4F</td>
<td>Yes</td>
<td>Attribute character</td>
</tr>
<tr>
<td>6F</td>
<td>No</td>
<td>Attribute character</td>
</tr>
</tbody>
</table>

n = value of byte
LOC OBJ LINE SOURCE STATEMENT

1 NAME FASTCO
2 CSSEG
3
4 ; FASTCO fills the CRT screen with the 2000 bytes of memory starting at BC.
5 ; All the bytes are assumed to be less than 80H.
6
7 FASTCO:

0000 C5 8 PUSH B ; save the pointer
0001 8002 9 MVI C,80H ; IOC block move to CRT command
0003 C02400 10 CALL PIOCIC ; put out the command byte
0006 110000 11 LXI D,0 ; coordinates for beginning of screen
0009 CD2F00 12 CALL PDEIOC ; put them out
000C 1109E7 13 LXI D,80*25 ; number of bytes being output
000F 81 14 POP H ; HL points to memory to be moved
0010 F1 15 FCLOOP:

0010 7A 16 MOV A,D ; count exhausted?
0011 B3 17 ORA E
0012 C1E500 18 JZ FCEXIT ; if so then all data is sent
0015 4E 19 MOV C,M ; put out one data byte
0016 23 20 INX H
0017 C03400 21 CALL PIOCIC
001A 1B 22 DCX D ; count down
001B C11000 23 JMP FCLOOP ; loop for next data byte

001C 8EFP 24 FCEXIT:

001E C0 25 MVI C,FFDH ; termination byte
0020 C03400 26 CALL PIOCIC ; send it to IOC
0023 C9 27 RET ; block movement to CRT is complete

30 ; PIOCIC sends the C register as a command byte to the IOC.
32
33 PIOCIC:

0024 DBC1 34 IN @C1H ; check status of IOC
0026 E687 35 ANI 11LB ; bottom three bits of status byte are examined
0028 C22400 36 JNZ PIOCIC ; they must be zero for IOC to be ready to receive
002B 79 37 MOV A,C ; IOC command byte
002C DBC1 38 OUT @C1H ; port Cl is the IOC command output port
002E C9 39 RET

40 ; PIOCIC sends data bytes E then D to the IOC.
42 ; PIOCIC sends the C register as a data byte to the IOC.
44
46 PDEIOC:

002F 4B 45 MOV C,E ; E is the first data byte; parameter to PIOCIC
0030 C03400 46 CALL PIOCIC ; send it
0033 4A 47 MOV C,D ; D is the second data byte
0034 DBC1 48 PIOCIC:

0036 E607 49 IN @C1H ; check status of IOC
0038 C23400 50 ANI 11LB ; bottom three bits of status byte are examined
003B 79 51 JNZ PIOCIC ; they must be zero for IOC to be ready to receive
003C DBC1 52 MOV A,C ; get byte to be output
003E C9 53 OUT @C1H ; port C9 is the IOC data output port

54 ; GDIOC fetches an IOC data byte into the A register.
56
58 GDIOC:

003F DBC1 59 IN @C1H ; check status of IOC
0041 E607 60 ANI 11LB ; bottom three bits of status byte are examined
0043 3B 61 DCW A ; they must be 01f for IOC to be ready to send
0044 C23F00 62 JNZ GDIOC
0047 DCW 63 IN @C0H ; port C0 is the input port for IOC data
0049 C9 64 RET

65
66 END

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

FASTCO C 0000 FCEXIT C 001E FCLOOP C 0019 GDIOC C 003F PIOCIC C 0024 PDEIOC C 002F PIOCIC C 0034

ASSEMBLY COMPLETE, NO ERRORS

Figure 4-1. Sample Program: FASTCO
This appendix lists all of the Intel keyboard characters and their hexadecimal values.

<table>
<thead>
<tr>
<th>Key</th>
<th>Unshift</th>
<th>Unshift</th>
<th>Shift</th>
<th>CNTL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TPWR Lock</td>
<td>TPWR Unlock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>61</td>
<td>41</td>
<td>41</td>
<td>01</td>
</tr>
<tr>
<td>B</td>
<td>62</td>
<td>42</td>
<td>42</td>
<td>02</td>
</tr>
<tr>
<td>C</td>
<td>63</td>
<td>43</td>
<td>43</td>
<td>03</td>
</tr>
<tr>
<td>D</td>
<td>64</td>
<td>44</td>
<td>44</td>
<td>04</td>
</tr>
<tr>
<td>E</td>
<td>65</td>
<td>45</td>
<td>45</td>
<td>05</td>
</tr>
<tr>
<td>F</td>
<td>66</td>
<td>46</td>
<td>46</td>
<td>06</td>
</tr>
<tr>
<td>G</td>
<td>67</td>
<td>47</td>
<td>46</td>
<td>07</td>
</tr>
<tr>
<td>H</td>
<td>68</td>
<td>48</td>
<td>48</td>
<td>08</td>
</tr>
<tr>
<td>I</td>
<td>69</td>
<td>49</td>
<td>49</td>
<td>09</td>
</tr>
<tr>
<td>J</td>
<td>6A</td>
<td>4A</td>
<td>4A</td>
<td>0A</td>
</tr>
<tr>
<td>K</td>
<td>6B</td>
<td>4B</td>
<td>4B</td>
<td>0B</td>
</tr>
<tr>
<td>L</td>
<td>6C</td>
<td>4C</td>
<td>4C</td>
<td>0C</td>
</tr>
<tr>
<td>M</td>
<td>6D</td>
<td>4D</td>
<td>4D</td>
<td>0D</td>
</tr>
<tr>
<td>N</td>
<td>6E</td>
<td>4E</td>
<td>4E</td>
<td>0E</td>
</tr>
<tr>
<td>O</td>
<td>6F</td>
<td>4F</td>
<td>4F</td>
<td>0F</td>
</tr>
<tr>
<td>P</td>
<td>70</td>
<td>50</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>Q</td>
<td>71</td>
<td>51</td>
<td>51</td>
<td>11</td>
</tr>
<tr>
<td>R</td>
<td>72</td>
<td>52</td>
<td>52</td>
<td>12</td>
</tr>
<tr>
<td>S</td>
<td>73</td>
<td>53</td>
<td>53</td>
<td>13</td>
</tr>
<tr>
<td>T</td>
<td>74</td>
<td>54</td>
<td>54</td>
<td>14</td>
</tr>
<tr>
<td>U</td>
<td>75</td>
<td>55</td>
<td>55</td>
<td>15</td>
</tr>
<tr>
<td>V</td>
<td>76</td>
<td>56</td>
<td>56</td>
<td>16</td>
</tr>
<tr>
<td>W</td>
<td>77</td>
<td>57</td>
<td>57</td>
<td>17</td>
</tr>
<tr>
<td>X</td>
<td>78</td>
<td>58</td>
<td>58</td>
<td>18</td>
</tr>
<tr>
<td>Y</td>
<td>79</td>
<td>59</td>
<td>59</td>
<td>19</td>
</tr>
<tr>
<td>Z</td>
<td>7A</td>
<td>5A</td>
<td>5A</td>
<td>1A</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>7E</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>39</td>
<td>39</td>
<td>39</td>
<td>29</td>
</tr>
<tr>
<td>-</td>
<td>2D</td>
<td>2D</td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>{</td>
<td>5B</td>
<td>5B</td>
<td>5B</td>
<td>7B</td>
</tr>
<tr>
<td>}</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>7D</td>
</tr>
<tr>
<td>\</td>
<td>5C</td>
<td>5C</td>
<td>5C</td>
<td>7C</td>
</tr>
<tr>
<td>@</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>#</td>
<td>5F</td>
<td>5F</td>
<td>5F</td>
<td>5E</td>
</tr>
<tr>
<td>$</td>
<td>3B</td>
<td>3B</td>
<td>3B</td>
<td>2B</td>
</tr>
<tr>
<td>%</td>
<td>3A</td>
<td>3A</td>
<td>3A</td>
<td>2A</td>
</tr>
<tr>
<td>&amp;</td>
<td>2C</td>
<td>2C</td>
<td>2C</td>
<td>3C</td>
</tr>
<tr>
<td>(</td>
<td>2E</td>
<td>2E</td>
<td>2E</td>
<td>3E</td>
</tr>
<tr>
<td>)</td>
<td>2F</td>
<td>2F</td>
<td>2F</td>
<td>3F</td>
</tr>
<tr>
<td>RETURN</td>
<td>0D</td>
<td>0D</td>
<td>0D</td>
<td></td>
</tr>
<tr>
<td>SPACE BAR</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>ESC</td>
<td>1B</td>
<td>1B</td>
<td>1B</td>
<td></td>
</tr>
<tr>
<td>RUBOUT</td>
<td>7F</td>
<td>7F</td>
<td>7F</td>
<td></td>
</tr>
<tr>
<td>↑</td>
<td>1E</td>
<td>1E</td>
<td>1E</td>
<td></td>
</tr>
<tr>
<td>↓</td>
<td>1C</td>
<td>1C</td>
<td>1C</td>
<td></td>
</tr>
<tr>
<td>←</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>→</td>
<td>1F</td>
<td>1F</td>
<td>1F</td>
<td></td>
</tr>
<tr>
<td>HOME</td>
<td>1D</td>
<td>1D</td>
<td>1D</td>
<td></td>
</tr>
</tbody>
</table>
**FUNC Key**

When the FUNC key is pressed simultaneously with any other key (or key combination), the code returned depends on whether the IOC is in local mode or on-line mode.

If the IOC is in local mode, 80H is added to the code that would have been returned if FUNC had not been pressed. The 80H bit is output to the console only if the USES8BIT$FLAG is set to true. (See Chapter 3, section “USES8BIT$FLAG.”)

If the IOC is in on-line mode, the preprogrammed sequence is returned as described in Chapter 2.

To put the IOC in local mode, type:

```
CNTL SHIFT L
```

(Hold down the CNTL key and the SHIFT key, then type L.)

To put the IOC in on-line mode, type:

```
CNTL SHIFT O
```
<table>
<thead>
<tr>
<th>ASCII Character</th>
<th>HEX</th>
<th>ASCII Character</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL</td>
<td>00</td>
<td>@</td>
<td>40</td>
</tr>
<tr>
<td>SOH</td>
<td>01</td>
<td>A</td>
<td>41</td>
</tr>
<tr>
<td>STX</td>
<td>02</td>
<td>B</td>
<td>42</td>
</tr>
<tr>
<td>ETX</td>
<td>03</td>
<td>C</td>
<td>43</td>
</tr>
<tr>
<td>EOT</td>
<td>04</td>
<td>D</td>
<td>44</td>
</tr>
<tr>
<td>ENQ</td>
<td>05</td>
<td>E</td>
<td>45</td>
</tr>
<tr>
<td>ACK</td>
<td>06</td>
<td>F</td>
<td>46</td>
</tr>
<tr>
<td>BEL</td>
<td>07</td>
<td>G</td>
<td>47</td>
</tr>
<tr>
<td>BS</td>
<td>08</td>
<td>H</td>
<td>48</td>
</tr>
<tr>
<td>HT</td>
<td>09</td>
<td>I</td>
<td>49</td>
</tr>
<tr>
<td>LF</td>
<td>0A</td>
<td>J</td>
<td>4A</td>
</tr>
<tr>
<td>VT</td>
<td>0B</td>
<td>K</td>
<td>4B</td>
</tr>
<tr>
<td>FF</td>
<td>0C</td>
<td>L</td>
<td>4C</td>
</tr>
<tr>
<td>CR</td>
<td>0D</td>
<td>M</td>
<td>4D</td>
</tr>
<tr>
<td>SO</td>
<td>0E</td>
<td>N</td>
<td>4E</td>
</tr>
<tr>
<td>SI</td>
<td>0F</td>
<td>O</td>
<td>4F</td>
</tr>
<tr>
<td>DLE</td>
<td>10</td>
<td>P</td>
<td>50</td>
</tr>
<tr>
<td>DC1</td>
<td>11</td>
<td>Q</td>
<td>51</td>
</tr>
<tr>
<td>DC2</td>
<td>12</td>
<td>R</td>
<td>52</td>
</tr>
<tr>
<td>DC3</td>
<td>13</td>
<td>S</td>
<td>53</td>
</tr>
<tr>
<td>DC4</td>
<td>14</td>
<td>T</td>
<td>54</td>
</tr>
<tr>
<td>NAK</td>
<td>15</td>
<td>U</td>
<td>55</td>
</tr>
<tr>
<td>SYN</td>
<td>16</td>
<td>V</td>
<td>56</td>
</tr>
<tr>
<td>ETB</td>
<td>17</td>
<td>W</td>
<td>57</td>
</tr>
<tr>
<td>CAN</td>
<td>18</td>
<td>X</td>
<td>58</td>
</tr>
<tr>
<td>EM</td>
<td>19</td>
<td>Y</td>
<td>59</td>
</tr>
<tr>
<td>SUB</td>
<td>1A</td>
<td>Z</td>
<td>5A</td>
</tr>
<tr>
<td>ESC</td>
<td>1B</td>
<td>^ (↑)</td>
<td>5B</td>
</tr>
<tr>
<td>FS</td>
<td>1C</td>
<td>\ ()</td>
<td>5C</td>
</tr>
<tr>
<td>GS</td>
<td>1D</td>
<td>]</td>
<td>5D</td>
</tr>
<tr>
<td>RS</td>
<td>1E</td>
<td>_</td>
<td>5F</td>
</tr>
<tr>
<td>US</td>
<td>1F</td>
<td>!</td>
<td>60</td>
</tr>
<tr>
<td>space</td>
<td>20</td>
<td>&quot;</td>
<td>61</td>
</tr>
<tr>
<td>!</td>
<td>21</td>
<td>a</td>
<td>62</td>
</tr>
<tr>
<td>#</td>
<td>22</td>
<td>b</td>
<td>63</td>
</tr>
<tr>
<td>$</td>
<td>23</td>
<td>c</td>
<td>64</td>
</tr>
<tr>
<td>%</td>
<td>24</td>
<td>d</td>
<td>65</td>
</tr>
<tr>
<td>&amp;</td>
<td>25</td>
<td>e</td>
<td>66</td>
</tr>
<tr>
<td>(</td>
<td>26</td>
<td>f</td>
<td>67</td>
</tr>
<tr>
<td>)</td>
<td>27</td>
<td>g</td>
<td>68</td>
</tr>
<tr>
<td>+</td>
<td>28</td>
<td>h</td>
<td>69</td>
</tr>
<tr>
<td>.</td>
<td>29</td>
<td>i</td>
<td>6A</td>
</tr>
<tr>
<td>/</td>
<td>2A</td>
<td>j</td>
<td>6B</td>
</tr>
<tr>
<td>0</td>
<td>2B</td>
<td>k</td>
<td>6C</td>
</tr>
<tr>
<td>1</td>
<td>2C</td>
<td>l</td>
<td>6D</td>
</tr>
<tr>
<td>2</td>
<td>2D</td>
<td>m</td>
<td>6E</td>
</tr>
<tr>
<td>3</td>
<td>2E</td>
<td>n</td>
<td>6F</td>
</tr>
<tr>
<td>4</td>
<td>2F</td>
<td>o</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>p</td>
<td>71</td>
</tr>
<tr>
<td>6</td>
<td>31</td>
<td>q</td>
<td>72</td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>r</td>
<td>73</td>
</tr>
<tr>
<td>8</td>
<td>33</td>
<td>s</td>
<td>74</td>
</tr>
<tr>
<td>9</td>
<td>34</td>
<td>t</td>
<td>75</td>
</tr>
<tr>
<td>;</td>
<td>35</td>
<td>u</td>
<td>76</td>
</tr>
<tr>
<td>&lt;</td>
<td>36</td>
<td>v</td>
<td>77</td>
</tr>
<tr>
<td>=</td>
<td>37</td>
<td>w</td>
<td>78</td>
</tr>
<tr>
<td>:</td>
<td>38</td>
<td>x</td>
<td>79</td>
</tr>
<tr>
<td>;</td>
<td>39</td>
<td>y</td>
<td>7A</td>
</tr>
<tr>
<td>&lt;</td>
<td>3A</td>
<td>z</td>
<td>7B</td>
</tr>
<tr>
<td>=</td>
<td>3B</td>
<td>{</td>
<td>7C</td>
</tr>
<tr>
<td>=</td>
<td>3C</td>
<td>}</td>
<td>7D</td>
</tr>
</tbody>
</table>
Within this index $f$ or $ff$ after a page number means and the following page (or pages).

attribute bytes, 3-3
attribute features, 3-3

block write commands, 4-1f
  0F, 4-1f
  2F, 4-1f
  4F, 4-1f
  6F, 4-1f
block write function, 4-1f

clear line, 3-1, 3-5
clear screen, 3-1, 3-4, 3-5
CNTL G, 3-5
CNTL J, 3-4
console output functions, 3-1ff
ESC A, 3-4
ESC B, 3-4
ESC C, 3-4
ESC D, 3-4
ESC E, 3-4
ESC H, 3-4
ESC J, 3-5
ESC K, 3-5
ESC R, 3-1
ESC S, 3-1
ESC T, 3-1
ESC W, 3-2
ESC X, 3-3f
ESC Y, 3-1

flags, 3-3f
FUNC key, 2-1, A-2

GDIOC, 4-1

HELP command, 2-1
HOME, 3-4

Input/Output Controller Board, 1-1
insert line function, 3-2
interface procedures, 4-1
GDIOC, 4-1
PCIOC, 4-1
PDIOC, 4-1

interrupt procedure, 4-2
interrupts, 4-2
IOC, 1-1, 2-1, 3-1, 3-4, 4-1f, A-2

keyboard codes, A-1
keys, 2-1
keywords, 2-2

left arrow, 3-4
local mode, A-2

on-line mode, A-2

PAUSE$FLAG, 3-3f
PCIOC, 4-1
PDIOC, 4-1
preprogrammed keys, 2-1f

repeat function, 2-2
RETURN, 3-5
right arrow, 3-4
RUBOUT, 3-5

up arrow, 3-4
USE$BIT$FLAG, 3-3
REQUEST FOR READER’S COMMENTS

Intel’s Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

2. Does the publication cover the information you expected or required? Please make suggestions for improvement.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

4. Did you have any difficulty understanding descriptions or wording? Where?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating).

NAME ________________________________ DATE __________________
TITLE ________________________________
COMPANY NAME/DEPARTMENT __________________________
ADDRESS ________________________________
CITY __________________ STATE __________ ZIP CODE __________
(COUNTRY)

Please check here if you require a written reply. ☐
WE'D LIKE YOUR COMMENTS ...

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

BUSINESS REPLY MAIL
FIRST CLASS    PERMIT NO. 1040    SANTA CLARA, CA
POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
Attn: Technical Publications M/S 6-2000
3065 Bowers Avenue
Santa Clara, CA 95051