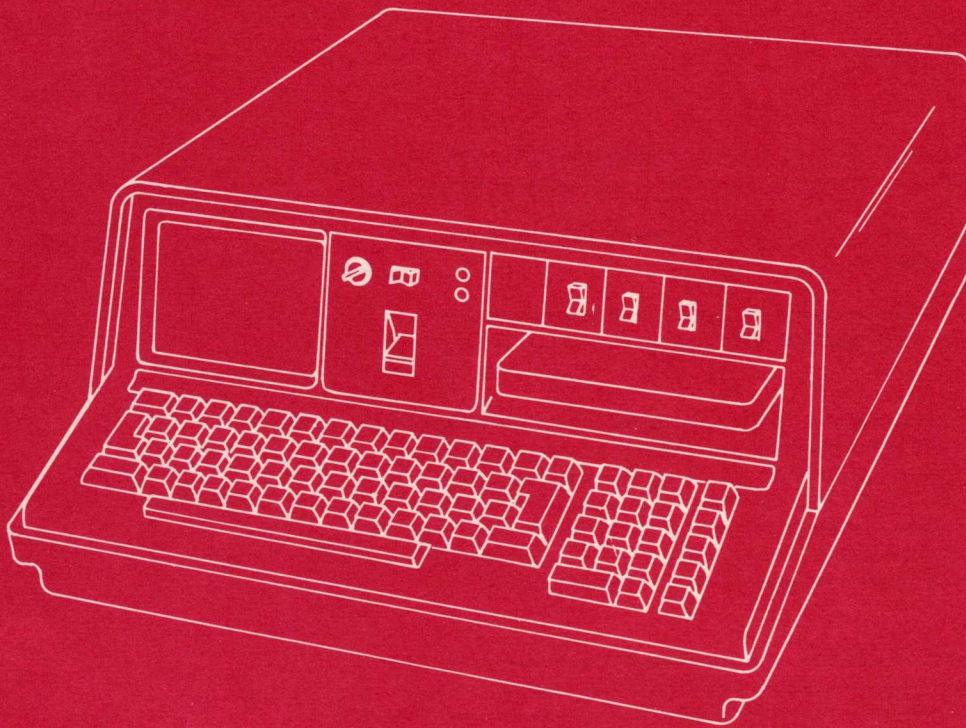




**IBM 5100**  
**Serial I/O Adapter Feature**  
**User's Manual**

**5100**



*IBM 5100*  
*Serial I/O Adapter Feature*  
*User's Manual*

## Preface

This manual is intended for users of the IBM 5100 Portable Computer with the Serial I/O Adapter feature. This manual assumes that you know how to operate the 5100 using the APL or the BASIC language, and that you are familiar with the characteristics of your I/O device.

### Related Publications

- *IBM 5100 BASIC Introduction*, SA21-9216
- *IBM 5100 APL Introduction*, SA21-9212
- *IBM 5100 BASIC Reference Manual*, SA21-9217
- *IBM 5100 APL Reference Manual*, SA21-9213

### Second Edition (January 1977)

This is a major revision of, and obsoletes, SA21-9239-0 and Technical Newsletter SN21-0260. Extensive changes have been made to this manual and it should be reviewed in its entirety.

Changes are continually made to the specifications herein; any such change will be reported in subsequent revisions or technical newsletters.

The document is not intended to be a performance or functional specification. It assumes that the user has an understanding of the interrelationship among any affected systems, machines, programs, or media.

IBM may have patents or pending patent applications covering subject matter described in this document, including appearance design patents or applications. The furnishing of this document does not constitute or imply a grant of any license under any patents, patent application, trademarks, copyrights or other rights of IBM or of any third party, or any right to refer to IBM in any advertising or other promotional or marketing activities. IBM assumes no responsibilities for any infringement of patents or other rights that may result from use of this document or from the manufacture, use, lease, or sale of apparatus described herein.

Licenses under IBM's utility patents are available on reasonable and nondiscriminatory terms and conditions. IBM does not grant licenses under its appearance design patents. Inquiries relative to licensing should be directed in writing to the Director of Contracts and Licensing, International Business Machines Corporation, Armonk, New York 10504.

A Reader's Comment Form is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. Comments become the property of IBM.



<b>CHAPTER 1. INTRODUCTION</b> . . . . .	<b>1</b>	<b>CHAPTER 4. USING THE APL LANGUAGE AND THE SERIAL I/O ADAPTER FEATURE</b> . . . . .	<b>42</b>
IBM 5100 Portable Computer Serial I/O Adapter		Output Operations . . . . .	42
Feature Overview . . . . .	1	Input Operations . . . . .	43
Things to Know About Your I/O Device . . . . .	2	End-of-File Conditions During APL Input Operations . . . . .	43
Getting Started . . . . .	3	Establishing a Shared Variable and Specifying the Device Characteristics . . . . .	44
Example Programs . . . . .	4	Opening the I/O Device . . . . .	45
APL Example Program . . . . .	4	Transferring Data . . . . .	46
BASIC Example Program . . . . .	5	Transmitting Data to the I/O Device (Output Operation) . . . . .	46
		Receiving Data From the I/O Device (Input Operation) . . . . .	46
<b>CHAPTER 2. HOW THE SERIAL I/O ADAPTER FEATURE WORKS.</b> . . . . .	<b>6</b>	Closing the I/O Device and Retracting the Shared Variable . . . . .	47
How the 5-, 6-, and 7-Bit Codes are Used . . . . .	6	5100 Characters that Require Using the Atomic Vector . . . . .	47
How the 8-Bit Codes are Used . . . . .	9	Input Data Flow When Using APL . . . . .	48
Input and Output Process Arrows . . . . .	12	Output Data Flow When Using APL . . . . .	49
Timing Requirements . . . . .	12	Sample APL Program . . . . .	50
Loading IMFs . . . . .	13	Using the Sample APL Program . . . . .	52
If the Serial I/O Adapter Program is Already Loaded . . . . .	13		
The Serial I/O Adapter Buffer . . . . .	14	<b>CHAPTER 5. USING THE BASIC LANGUAGE AND THE SERIAL I/O ADAPTER FEATURE</b> . . . . .	<b>53</b>
Input New-Line and Input End-of-Block Characters . . . . .	16	Output Operations . . . . .	53
Input End-of-Record and End-of-Buffer Conditions . . . . .	17	Input Operations . . . . .	54
Out New-Line and Output End-of-Blocks Characters . . . . .	19	End-of-File Conditions During BASIC Input Operations . . . . .	54
Output End-of-Record and End-of-Buffer Conditions . . . . .	20	Specifying the Device Characteristics . . . . .	55
Opening the Command and I/O Devices . . . . .	22	Opening the I/O Device for Input or Output Operations . . . . .	56
		Performing Input or Output Operations . . . . .	57
<b>CHAPTER 3. USER-SPECIFIED DEVICE CHARACTERISTICS.</b> . . . . .	<b>24</b>	Closing the Input or Output Device . . . . .	57
Set Input Buffer Size . . . . .	25	Loading, Listing, or Punching a BASIC Program . . . . .	58
Set Output Buffer Size . . . . .	25	Special Characters . . . . .	59
Set Data Rate . . . . .	26	For a BASIC-Only Keyboard . . . . .	59
Set Character or Numeric Mode . . . . .	27	5100 Characters That are Not Shown on the Keyboard . . . . .	62
Set Prompting or Nonprompting Mode . . . . .	28	Input Data Flow When Using BASIC . . . . .	63
Set Input Prompting Character . . . . .	29	Output Data Flow When Using BASIC . . . . .	64
Set Modem, Ignore, Terminal, Wait, or Set Mode . . . . .	30	Sample BASIC Program . . . . .	65
Set Receive Timeout . . . . .	32	Using the Sample BASIC Program . . . . .	67
Set Input New-Line Character . . . . .	33		
Set Input End-of-Block Character . . . . .	33	<b>CHAPTER 6. SERIAL I/O ADAPTER CHARACTERISTICS AND SPECIFICATIONS.</b> . . . . .	<b>69</b>
Set Output New-Line Character . . . . .	33	Connector and Pin Assignments . . . . .	70
Set Output End-of-Block Character . . . . .	33		
Set Enable or Disable Input New-Line and Input End-of-Block Characters . . . . .	34	<b>APPENDIX A. SET-UP PROCEDURE</b> . . . . .	<b>71</b>
Set Enable or Disable Output New-Line and Output End-of-Block Characters . . . . .	35	Attaching the External I/O Device . . . . .	72
Set Immediate Delay . . . . .	36	Serial I/O Adapter Self Test . . . . .	74
Set APL or BASIC Translation Table . . . . .	36		
Change the Output Translation Table . . . . .	37	<b>APPENDIX B. 5-BIT CODE CONVERSION CHART</b> . . . . .	<b>77</b>
Change the Input Translation Table . . . . .	38	Notes About Letters (Lower Shift) and Figures (Upper Shift) . . . . .	79
Set Input/Output Parity . . . . .	39		
Set Number of Stop Bits . . . . .	39		
User-Specified Device Characteristic and Default Value Summary . . . . .	40		

<b>APPENDIX C. 6-BIT CODE CONVERSION CHART . . . . .</b>	<b>80</b>
Notes About Lower Shift and Upper Shift. . . . .	83
<b>APPENDIX D. 7-BIT CODE CONVERSION CHART. . . . .</b>	<b>84</b>
<b>APPENDIX E. DEVICE ADDRESS AND USER-</b>	
<b>    SPECIFIED DEVICE CHARACTERISTIC SUMMARY . . . . .</b>	<b>88</b>
Device Addresses . . . . .	88
User-Specified Device Characteristics. . . . .	89
<b>APPENDIX F. ERROR MESSAGES. . . . .</b>	<b>91</b>
<b>GLOSSARY. . . . .</b>	<b>96</b>
<b>INDEX. . . . .</b>	<b>98</b>

### IBM 5100 PORTABLE COMPUTER SERIAL I/O ADAPTER FEATURE OVERVIEW

The IBM 5100 Portable Computer Serial I/O Adapter feature allows a variety of external I/O devices to be attached to an IBM 5100, such as:

- Plotters and graphic display devices
- Card readers and/or punches
- Printers
- Instrumentation devices

Only one external I/O device at a time can be attached to the 5100 and the device must supply its own power. The specifications for attaching an external I/O device using the Serial I/O Adapter feature are described in Chapter 4.

The Serial I/O Adapter feature is the connection between the 5100 language (APL or BASIC) and the external I/O device. The feature uses one of the following codes when transmitting data to or receiving data from the external I/O device:

- 5-bit
- 6-bit (plus parity)
- 7-bit (plus parity)
- 8-bit

The 5-bit, 6-bit, and 7-bit codes are compatible with I/O devices that use the Baudot, EBCD (extended binary coded decimal), and ASCII (American standard code for information interchange) character formats, respectively. The 8-bit code is compatible with I/O devices that use an 8-bit binary character format.

When transmitting data, the 5100 characters are converted into 5-bit, 6-bit, 7-bit, or 8-bit code. When receiving data, the 5-bit, 6-bit, 7-bit, or 8-bit codes are converted into 5100 characters.

Because the Serial I/O Adapter feature interacts with the BASIC or APL language, it cannot be used while the communications feature (if installed) is being used.

## THINGS TO KNOW ABOUT YOUR I/O DEVICE

You should know the answers to the following questions before using the Serial I/O Adapter feature. The answers to these questions will help you to specify the correct device characteristics which are discussed in Chapter 3.

1. What code (5-bit, 6-bit, 7-bit, or 8-bit) does the I/O device use? \_\_\_\_\_
2. What is the data rate (bits per second) required by the I/O device? \_\_\_\_\_
3. Does the I/O device require an input prompting character from the 5100 to tell it to send data? \_\_\_\_\_
4. What cable (modem or terminal) is used to attach the I/O device to the 5100?  
\_\_\_\_\_

*Note:* See *Appendix A. Set Up Procedure* for information on which cable is used.

5. For 7- and 8-bit codes only, how many stop bits (one or two) are required by the I/O device (transmit operations only)? \_\_\_\_\_
6. For 7-bit code only, what parity (odd, even, or none) is required by the I/O device? \_\_\_\_\_

## GETTING STARTED

If the Serial I/O Adapter feature is installed, the following procedure allows you to use your 5100 with an external I/O device:

1. Insert the tape cartridge containing the Serial I/O Adapter program into the built-in tape unit and execute a )MODE COM command (if you are using APL) or UNTIL MODE COM command (if you are using BASIC) to display the option menu.
2. When the option menu is displayed, enter one of the following numbers or T to select the required code:
  - 5 (selects 5-bit code)
  - 6 (selects 6-bit code)
  - 7 (selects 7-bit code)
  - 8 (selects 8-bit code)
  - T (selects the Serial I/O Adapter self test; see *Serial I/O Adapter Self Test*, in Appendix A, for a description of this test)

Once the code is selected, the correct version of the Serial I/O Adapter program is loaded into user storage.

*Note:* Any user-defined programs or data is cleared from the user storage when the Serial I/O Adapter program is loaded. Also, the storage required for the Serial I/O Adapter program (approximately 4K bytes) is subtracted from the available user storage.

3. Now, using the 5100 language (APL or BASIC), you can:
  - a. Adapt the 5100 to the requirements of the I/O device by specifying the appropriate device characteristics (opening the command device and specifying the device characteristics are discussed in Chapters 2 and 3, respectively).
  - b. Open the I/O device and perform input or output operations (opening the I/O device is discussed in Chapter 2). When using APL, input or output operations are performed using an APL shared variable. When using BASIC, input or output operations are performed using GET, PUT, or PRINT statements or LOAD and SAVE commands.

## EXAMPLE PROGRAMS

The APL and BASIC example programs show how 80-column records might be read from a hypothetical I/O device. These example programs do not explain in detail how the Serial I/O Adapter feature works; however, they will give you an idea of how the Serial I/O Adapter feature interacts with the APL or BASIC languages. Chapters 2 and 3 describe how the Serial I/O Adapter feature works and how to specify the device characteristics. Chapters 4 and 5 describe using the Serial I/O Adapter feature with the APL or BASIC language, respectively.

### APL Example Program

```

V READC[]JW
V READ; DATA; X
[1]  A *****
[2]  A*  ESTABLISH A SHARED VARIABLE
[3]  A *****
[4]  A
[5]  1 []SVO 'DATA'
[6]  A
[7]  A *****
[8]  A*  ADAPT THE 5100 TO THE I/O DEVICE REQUIREMENTS
[9]  A *****
[10] A
[11] DATA←'OUT 31001'
[12] DATA←'R/300,H/P,I/82'
[13] DATA←10
[14] A
[15] A
[16] A
[17] A
[18] A
[19] A
[20] A
[21] A *****
[22] A  TRANSFER DATA FROM THE I/O DEVICE TO THE 5100
[23] A *****
[24] A
[25] A
[26] A
[27] DATA←'IN 33001'
[28] LOOP:X←DATA
[29] X
[30] →(✓/'END'≠3↑X)/LOOP
[31] DATA←10
[32] A
[33] A
[34] A
[35] A *****
[36] A*  RETRACT THE SHARED VARIABLE
[37] A *****
[38] A
[39] []SVR 'DATA'
V

```

Open the command device.

User-specified device characteristics; these device characteristics are used to adapt the 5100 to the I/O device requirements, such as, data rate, input prompting character, buffer size, and so on. Chapter 3 describes each user specified device characteristic in detail.

Close the command device (by assigning 10 to the shared variable).

Open the input device.

Assign an input record to the shared variable DATA. The value of DATA is then assigned to the variable X.

Display the input record.

This program expects the last input record to begin with the characters END.

Close the input device after the last input record is received.



## BASIC Example Program

```
0010 REM *****
0020 REM * ADAPT THE 5100 TO THE I/O DEVICE REQUIREMENTS *
0030 REM *****
0040 REM
0050 OPEN FL1, 'A08', OUT
0060 PRINT FL1, 'R/300,H/P,F/C,I/97'
0070 CLOSE FL1
0080 REM
0090 REM
0100 REM
0110 REM
0120 REM
0130 REM
0140 REM
0150 REM *****
0160 REM * OPEN THE I/O DEVICE FOR INPUT OPERATIONS *
0170 REM *****
0180 REM
0190 OPEN FL2, 'A02', IN
0200 REM
0210 REM *****
0220 REM * PERFORM I/O OPERATIONS *
0230 REM *****
0240 REM
0250 REM
0260 GET FL2, A$, B$, C$, D$, E$
0270 PRINT A$; B$; C$; D$; E$
0280 IF A$# 'END' GOTO 0260
0290 REM
0300 REM *****
0310 REM * CLOSE THE INPUT DEVICE AND END THE PROGRAM *
0320 REM *****
0330 REM
0340 CLOSE FL2
0350 END
```

Open the command device.

User-specified device characteristics; these device characteristics are used to adapt the 5100 to the I/O device requirements, such as, data rate, input prompting character, buffer size, and so on. Chapter 3 describes each user-specified device characteristic in detail.

Assign the input record to the referenced variables.

Display the input record.

This program expects the last input record to begin with the characters END.

## Chapter 2. How The Serial I/O Adapter Feature Works

The following sections contain information about:

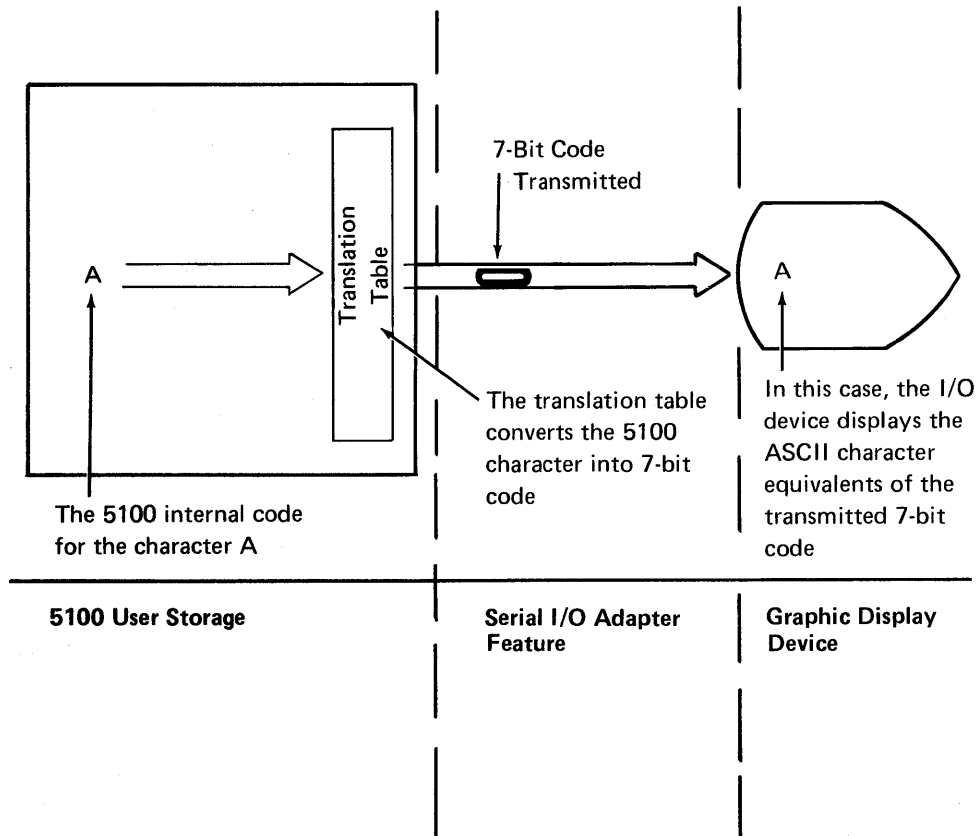
- The 5-, 6-, 7-, and 8-bit codes
- The input and output process arrows
- The timing requirements
- Loading IMFs (internal machine fixes)
- The buffers
- The input new-line and input end-of-block characters
- The output new-line and output end-of-block characters
- The device addresses (opening the command and I/O devices)

You should be familiar with this information when using the Serial I/O Adapter feature.

### HOW THE 5-, 6-, AND 7-BIT CODES ARE USED

The 5-bit, 6-bit, or 7-bit codes can represent Baudot, EBCD, or ASCII characters, respectively. However, internally the 5100 does not use these codes (characters). For output operations, the 5100 characters are converted into 5-, 6-, or 7-bit codes that are then transmitted by the 5100. For input operations, the 5-, 6-, or 7-bit codes received by the 5100 are converted to 5100 characters.

This conversion is done using the translation tables provided by the Serial I/O Adapter program. The translation table contains the 5100 character and the equivalent 5-, 6-, or 7-bit code. The following illustration shows a 5100 character converted to 7-bit code and transmitted to a graphic display device.



When the 5100 receives codes that contain parity or stop bit errors, these codes are converted into **␣** (O-U-T) characters. When transmitting data, untranslatable characters are transmitted as blanks (5- and 6-bit codes) or as all one bits (7-bit code).

#### **5-Bit Code Translation Tables**

There are two 5-bit code translation tables, one for input and one for output. The input table specifies the 5100 character to use for each 5-bit code received, and the output table specifies the 5-bit code to transmit for each 5100 character.

The 5-bit code input and output translation tables can be modified. See *Change The Output Translation Table* and *Change The Input Translation Table* in Chapter 3 for information on how to change these tables.

#### **6-Bit Code Translation Tables**

There are four 6-bit code translation tables, an input and an output table for the APL language and an input and an output table for the BASIC language. Depending upon which translation table (APL or BASIC) is specified, the input table specifies the 5100 character to use for each 6-bit code received, and the output table specifies the 6-bit code to transmit for each 5100 character.

The 6-bit code input and output translation tables can be modified. See *Change The Output Translation Table* and *Change The Input Translation Table* in Chapter 3 for information on how to change these tables.

#### **7-Bit Code Translation Tables**

There are two 7-bit code translation tables, one for input and one for output. The input table specifies the 5100 character to use for each 7-bit code received, and the output table specifies the 7-bit code to transmit for each 5100 character.

The 7-bit code input and output translation tables can be modified. See *Change The Output Translation Table* and *Change The Input Translation Table* in Chapter 3 for information on how to change these tables.

## HOW THE 8-BIT CODES ARE USED

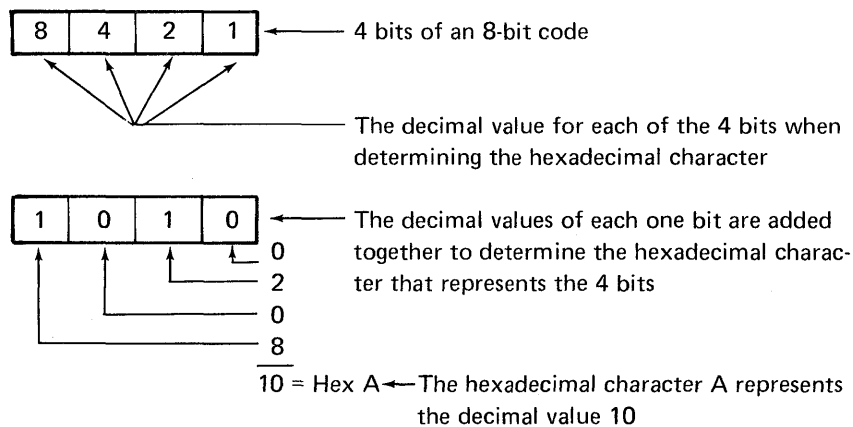
The 8-bit codes can represent 256 different binary values. To generate all 256 combinations, you must use the hexadecimal representation of the 8-bit codes.

*Note for BASIC users:* Hexadecimal constants (for example, A\$ = X'F9') are not used to enter the hexadecimal representation.

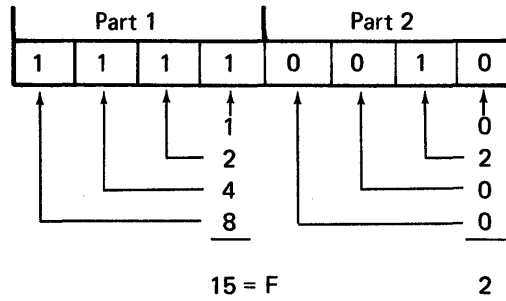
The hexadecimal representation of the 8-bit code is two characters from 0 through 9 or A through F. The decimal values and binary values for the characters A through F are:

Character	Decimal Value	Binary Value
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

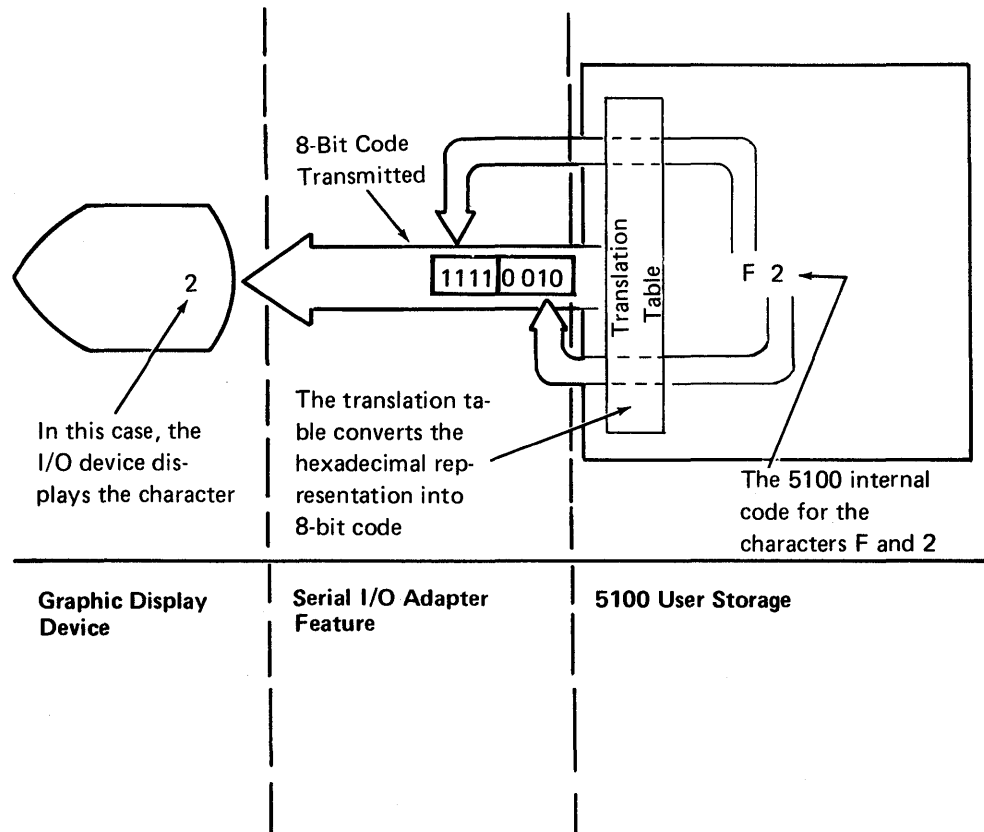
The 8-bit code is divided into two 4-bit parts, with each part represented by a single hexadecimal character. The hexadecimal character is determined as follows:



The following illustration shows an 8-bit code with a hexadecimal representation of F2:

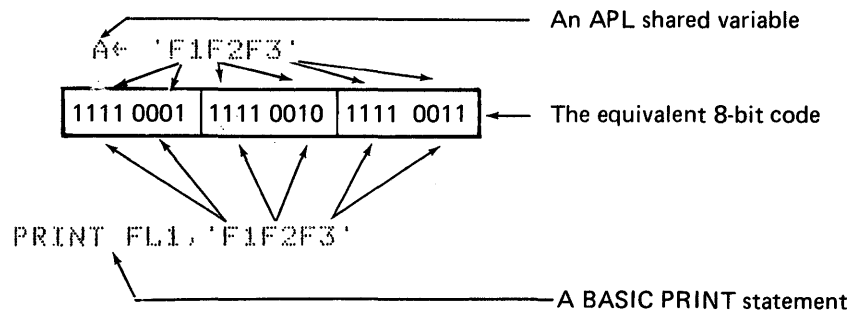


When transmitting data, the 5100 converts two characters (which must be the hexadecimal representation of 8 bits) into the 8-bit code. When receiving data, the 5100 converts the 8-bit code into two characters (which are the hexadecimal representation of the 8 bits). This conversion is done using a translation table provided by the Serial I/O Adapter program. The translation table contains the two-character hexadecimal representation of the 8-bit code. The following illustration shows two 5100 characters converted to 8-bit code and transmitted to a graphic display device (in this case, the I/O device recognizes this 8-bit code as the character 2):





The following example shows how the hexadecimal representation of an 8-bit code is entered using the APL and BASIC languages:



*Note:* A BASIC PUT statement should not be used when using 8-bit code.

**Notes:**

1. When transmitting data, two 5100 characters are converted to an 8-bit code, unless:
  - a. One of the characters is not a valid hexadecimal character (0-F). In this case, an 8-bit code of all ones (hex FF) is transmitted.
  - b. There is an odd number of characters in the record to be transmitted. In this case, an 8-bit code of all ones (hex FF) is transmitted for the last (odd) character in the record.
2. When receiving data, each 8-bit code is converted into two 5100 characters, unless:
  - a. The 8-bit code has an error (for example, stop bit error). In this case, the 8 bit code is converted into two **U** (O-U-T) symbols.
  - b. The 8-bit code represents an input new-line or input end-of-block character. See *Input New-Line and Input End-of-Block Characters* in Chapter 2 for more information about what happens when an input new-line or input end-of-block character is received.

## INPUT AND OUTPUT PROCESS ARROWS

During input or output operations, one of four arrows (←→ ↑ ↓) is displayed to indicate the status of the input or output operation. When using the APL language, the arrows are displayed in the upper right corner of the display screen. When using the BASIC language, the arrows are displayed in the lower right corner of the display screen. These arrows indicate the status of the input or output operation as follows:

Arrow	Meaning
←→	The 5100 is receiving data from the input device. This arrow moves in a back-and-forth motion as characters are received.
→	The 5100 is transmitting data to the output device. This arrow moves in a back-and-forth motion as the characters are transmitted.
↑	If in modem mode, the 5100 is waiting for the input device to request to send data (turn on request to send [RTS]). If in terminal mode, the 5100 is waiting for the output device to indicate it is ready for the transmitted data (turn on clear to send [CTS]).
↓	If in modem mode, the 5100 is waiting for the input device to stop requesting to send data (turn off request to send [RTS]). If in terminal mode, the 5100 is waiting for the output device to stop indicating it is ready for the transmitted data (turn off clear to send [CTS]).

*Note:* These arrows are not displayed if the IN PROCESS light is on.

## TIMING REQUIREMENTS

The user has the responsibility for determining that the timing requirements of the external I/O device can be met when other IBM 5100 input and/or output operations or computations are interleaved with the I/O device operation. Typical input or output operations take 300-600 milliseconds besides the actual data transmission time. Transmitting and receiving data through the Serial I/O Adapter feature cannot be done simultaneously or overlapped with other 5100 operations.

## LOADING IMFs

If IMFs (internal machine fixes) are required when using the Serial I/O Adapter feature, the IMFs must be loaded after the Serial I/O Adapter program is loaded. If you attempt to load the Serial I/O Adapter program and IMFs are already loaded, the program will not load and the message **IMFS LOADED PRESS EXECUTE** is displayed. In this case, you can do one of the following:

1. Press **RESTART**, load the Serial I/O Adapter program, and then load the required IMFs.
2. Press **EXECUTE**. The 5100 will return to the APL or BASIC language.

See *IBM 5100 APL Reference Manual*, SA21-9213, or *IBM 5100 BASIC Reference Manual*, SA21-9217, for information on how to load IMFs.

## IF THE SERIAL I/O ADAPTER PROGRAM IS ALREADY LOADED

If you attempt to load the Serial I/O Adapter program and one of the four versions of the Serial I/O Adapter program is already loaded, the program will not load (that is, the version of the program already loaded will not be replaced) and the following message is displayed:

ALREADY LOADED  
PRESS    T FOR TEST OPTION  
         EXECUTE FOR RETURN

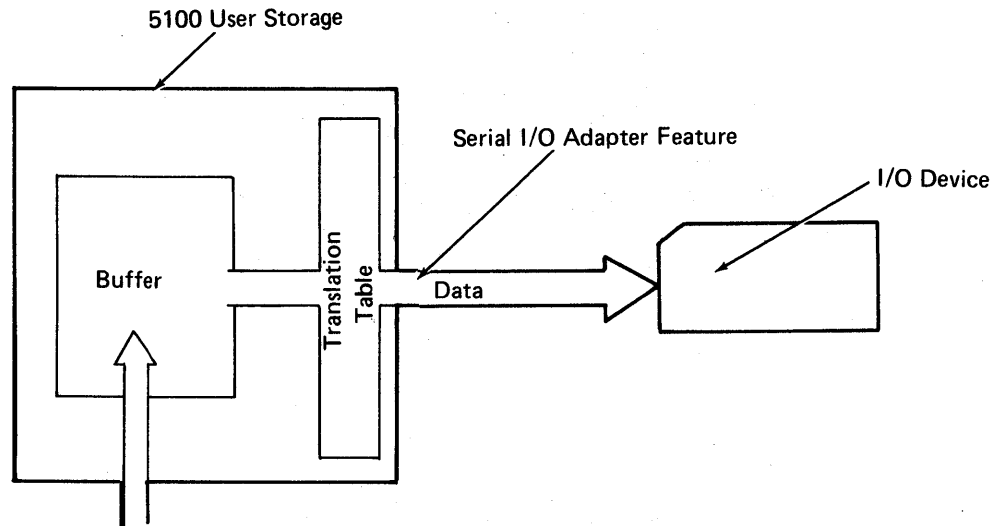
In this case, you can do one of the following:

1. Press **EXECUTE**. The 5100 will return to the APL or BASIC language and you can use the version of the Serial I/O Adapter program already loaded.
2. Press **RESTART** and load another version of the Serial I/O Adapter program.
3. Press **T** to select the Serial I/O Adapter self test.

## THE SERIAL I/O ADAPTER BUFFER

The serial I/O adapter buffer is part of the 5100 user storage and is reserved for input and output operations. During output operations, records are placed in the buffer until the data is transmitted. During input operations, records are received by the 5100 and placed in the buffer. After the 5100 is through receiving data, it then uses the data in the buffer. The following illustrations show the flow of data for input and output operations:

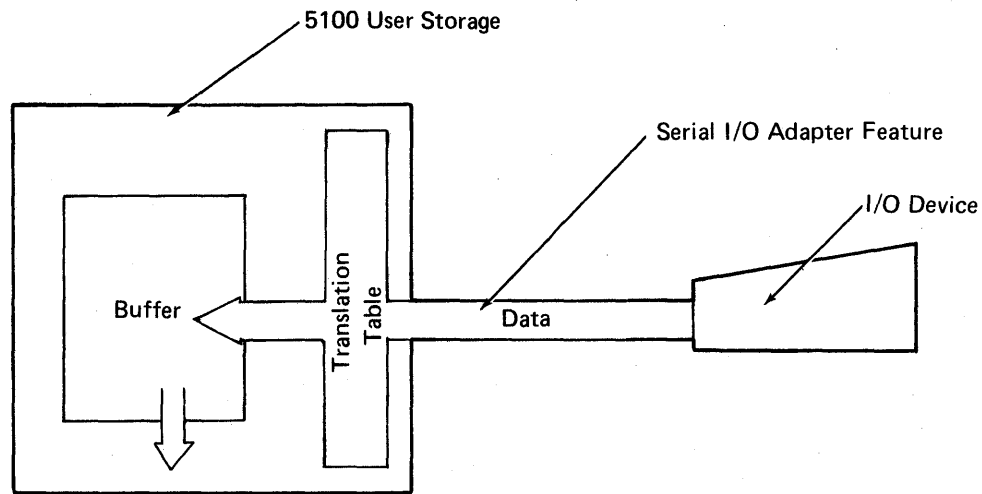
### Output Operations



Data is placed in the buffer using:

- An APL shared variable
- A BASIC PRINT or PUT statement
- A BASIC SAVE command

## Input Operations



Data is removed from the buffer using:

- An APL shared variable
- A BASIC GET statement
- A BASIC LOAD command

The size of the buffer determines how much data can be placed in the buffer at any given time. When the device characteristics (see Chapter 3) are being specified, the Serial I/O Adapter feature always sets the buffer size to 204 bytes. The buffer size for input or output operations can be specified (see *Set Input Buffer Size* or *Set Output Buffer Size* in Chapter 3); however, the Serial I/O Adapter feature always allocates at least 204 bytes for APL output operations and for BASIC SAVE operations. The maximum buffer size depends on the amount of available user storage.

**Note:** When using 8-bit code (see *How the 8-Bit Codes Are Used* in this chapter), two bytes of the buffer are required for each 8-bit code transmitted or received.

## INPUT NEW-LINE AND INPUT END-OF-BLOCK CHARACTERS

The input new-line and input end-of-block characters are characters (representing 5-, 6-, 7- or 8-bit codes) that have a special meaning when received by the 5100. These characters are used by the Serial I/O Adapter feature as follows:

- The input new-line character indicates that a complete record was received. This character is translated into the 5100 code for end of record and placed in the buffer.
- The input end-of-block character indicates that a complete block (one or more records) of data was received. This character is translated into the 5100 code for end of buffer and placed in the buffer.

If the input new-line and input end-of-block character both represent the same 5-, 6-, 7-, or 8-bit code, this character is translated into both 5100 codes for end of record and end of buffer when it is received.

Unless you specify otherwise, the 5100 character equivalent of the input new-line and input end-of-block characters are:

Code	Input New Line	Input End of Block
5-bit	< (carriage return)	<
6-bit	⌘ (end of transmission)	⌘
7-bit	< (carriage return)	<
8-bit	8D (carriage return)	8D

*Note:* If you are using the BASIC language, the 5100 characters not shown on the keyboard are described under *Special Characters* in Chapter 5.

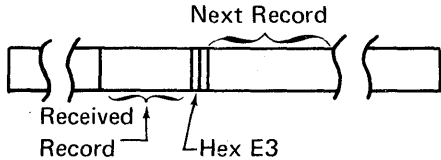
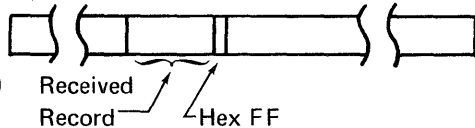
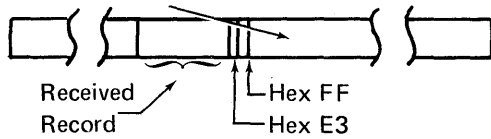
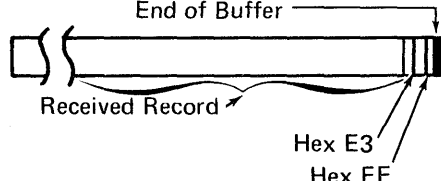
See *Set Input New-Line Character* and *Set Input End-of-Block Character* in Chapter 3 for information on how to specify input new-line and input end-of-block characters.



## INPUT END-OF-RECORD AND END-OF-BUFFER CONDITIONS

The 5100 takes the following actions when an input new-line character, input end-of-block character, or the physical end of buffer is encountered when receiving data.

*Note:* See *Set Enable/Disable Input New-Line and Input End-of-Block Characters* in Chapter 3 for information on disabling the actions normally taken by the 5100 when an input new-line or input end-of-block character is received.

Condition	Action Taken	Graphic Representation of the Buffer
An input new-line character is received (indicating end of record).	The 5100 code for end of record (hex E3) is placed in the buffer and the 5100 continues receiving data.	 <p>The diagram shows a buffer divided into segments. The first segment is labeled 'Received Record'. The second segment contains 'Hex E3'. The third segment is labeled 'Next Record'.</p>
An input end-of-block character is received (indicating end of data).	The 5100 code for end of buffer (hex FF) is placed in the buffer and the 5100 continues receiving data.	 <p>The diagram shows a buffer divided into segments. The first segment is labeled 'Received Record'. The second segment contains 'Hex FF'.</p>
<p><i>Note:</i> If the input prompting mode is specified, the 5100 sends an input prompting character after the hex FF is placed in the buffer.</p>		
A character that represents both an input new-line and input end-of-block character is received.	The 5100 codes for end of record and end of buffer are placed in the buffer and the 5100 stops receiving data.	<p>The remaining portion of the buffer is unused.</p>  <p>The diagram shows a buffer divided into segments. The first segment is labeled 'Received Record'. The second segment contains 'Hex FF'. The third segment contains 'Hex E3'.</p>
Only two positions remain unused in the buffer.	The 5100 codes for end of record and end of buffer are placed in the last two positions of the buffer and the 5100 stops receiving data.	 <p>The diagram shows a buffer divided into segments. The first segment is labeled 'Received Record'. The last two segments contain 'Hex E3' and 'Hex FF' respectively. The label 'End of Buffer' points to the end of the buffer.</p>

The following conditions apply to BASIC only:

**Condition**

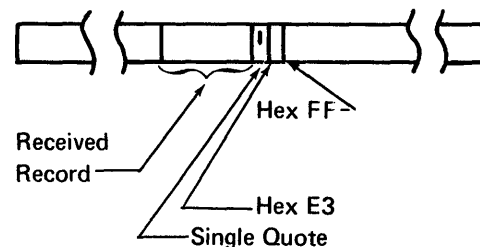
**Action Taken**

**Graphic Representation  
of the Buffer**

Character mode (F/C, see *Set Character or Numeric Mode* in Chapter 3) is being used and one of the following conditions occurs:

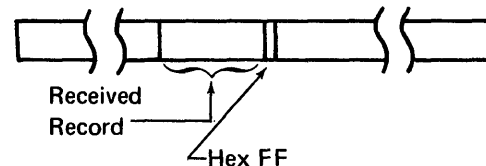
- An input new-line character is received
- Only three positions remain unused in the buffer

A single quote is placed in the buffer, followed by the 5100 codes for end of record and end of buffer, and the 5100 stops receiving data.



An input end-of-block character is received and the input new-line and input end-of-block characters are not specified as the same character

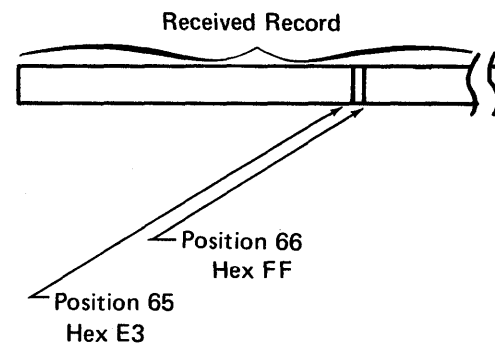
The 5100 code for end of buffer is placed in the buffer (without the single quote or end of record code) and the 5100 stops receiving data.



*Note:* This will cause an error condition; therefore, an input end-of-block character that is not equal to the input new-line character should not be used when the character mode (F/C) is used.

A BASIC source program is being loaded from an input device and an input new-line character is received.

The 5100 codes for end of record and end of buffer are placed in the buffer in positions 65 and 66, because BASIC source programs are limited to records equal to or less than 64 characters.



## OUTPUT NEW-LINE AND OUTPUT END-OF-BLOCK CHARACTERS

The output new-line and output end-of-block characters are characters (representing 5-, 6-, 7-, or 8-bit codes) that normally have a special meaning when transmitted to an output device (for example, to indicate a carriage return or line feed). These characters are transmitted when the following conditions occur:

1. Both an output new-line character and an output end-of-block character are transmitted when the 5100 code for end of record (hex E3) is encountered in the buffer.
2. An output new-line character or output end-of-block character is transmitted when the equivalent user-specified 5100 character is encountered in the buffer.
3. For BASIC only, if no output new-line or output end-of-block character was previously transmitted and the physical end of buffer is reached, output new-line and output end-of-block characters are transmitted. However, if one or more of these characters was previously transmitted, they are not transmitted when the physical end of buffer is reached.

**Note:** For conditions 1 and 3, if the output new-line and output end-of-block characters are both represented by the same character, only one character is transmitted to the output device. This transmitted character represents both a new-line and end-of-block condition.

Unless you specify otherwise, the 5100 character equivalent of the output new-line and output end-of-block characters are:

Code	Output New Line	Output End of Block
5-bit	C (carriage return)	n (line feed)
6-bit	I (new line)	I (new line)
7-bit	C (carriage return)	n (line feed)
8-bit	8D (carriage return)	0A (line feed)

**Note:** If you are using the BASIC language, the 5100 characters not shown on the keyboard are described under *Special Characters* in Chapter 5.

See *Set Output New-Line Character* and *Set Output End-of-Block Character* in Chapter 3 for information on how to specify output new-line and output end-of-block characters.

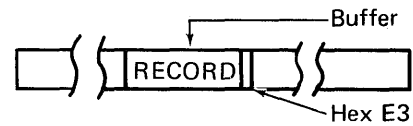
## OUTPUT END-OF-RECORD AND END-OF-BUFFER CONDITIONS

The 5100 places end-of-record (hex E3) and end-of-buffer (hex FF) codes in the buffer for the following conditions:

*For APL Operations:*

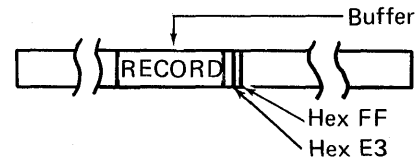
- An end-of-record code is placed after each record assigned to a shared variable.

```
SHAREDAVARIABLE ← 'RECORD'
```



- An end-of-buffer code is placed in the buffer when an empty vector is assigned to the shared variable.

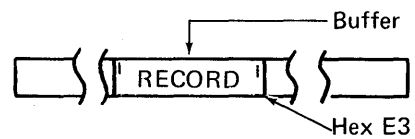
```
SHAREDAVARIABLE ← {}
```



*For BASIC Operations:*

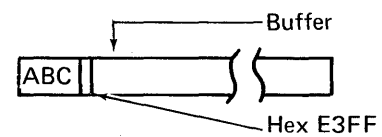
- An end-of-record code is placed after each record generated with a PUT statement.

```
PUT FL1, 'RECORD'
```



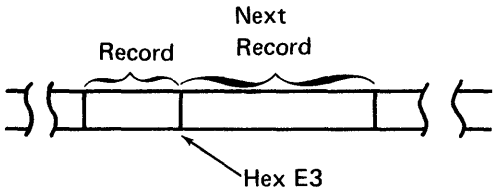
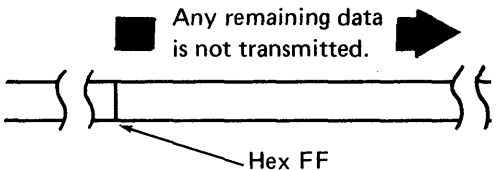
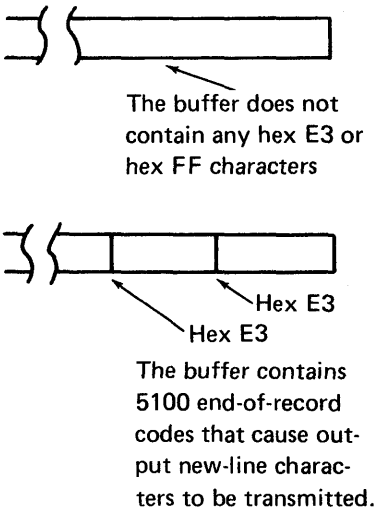
- A PRINT statement contains the end-of-record and/or end-of-buffer codes.

```
A$ = X'E3FF'  
PRINT FL1, 'ABC'; A$
```



The 5100 takes the following actions when an end-of-record code, end-of-buffer code, or physical end of buffer is encountered when transmitting data:

*Note:* See *Set Enable/Disable Output New-Line and Output End-of-Block Characters* in Chapter 3 for information on disabling the actions normally taken by the 5100 when an end-of-record code (hex E3) or end-of-buffer code (hex FF) is encountered when the 5100 is transmitting data.

Condition	Action Taken	Graphic Representation of the Buffer
The 5100 code for end of record is encountered.	Output new-line and output end-of-block characters are transmitted before the 5100 transmits the next record.	
The 5100 code for end of buffer is encountered.	The 5100 stops transmitting (without transmitting an output new-line or output end-of-block character). Any remaining data in the buffer is not transmitted.	
The physical end of the buffer is reached.	<p>For APL operations, the 5100 stops transmitting (without transmitting an output new-line or output end-of-block character).</p> <p>For BASIC operations, if no output new-line or output end-of-block characters were previously transmitted, these characters are transmitted and the 5100 stops transmitting. However, if output new-line and/or output end-of-block characters were previously transmitted, the 5100 stops transmitting (without transmitting an additional output new-line or output end-of-block character).</p>	

*Note:* If the output new-line and output end-of-block characters are both represented by the same character, only one character is transmitted to the output device. This transmitted character represents both a new-line and end-of-block condition.

## OPENING THE COMMAND AND I/O DEVICES

The Serial I/O Adapter feature requires special device addresses to open the command and I/O devices before any input or output operations can be performed. Following is an example of opening the command device before specifying the device characteristics (see Chapter 3) using the APL and the BASIC languages:

In APL, a shared variable must be used when opening a device.

```
A←'OUT (31)001 TYPE=I'
```

Command Device Address

```
OPEN FL1, 'A08', OUT
```

In BASIC, an OPEN statement must be used when opening a device.

*Note:* The command device is not an external hardware device; instead, the command device address refers to a part of the 5100 user storage where the values of the specified device characteristics are stored.

See Chapters 4 and 5 for more examples of opening the devices using the APL or BASIC languages, respectively.

### For APL

Device Address	Description	Use
31	Command device	Specifying the user device characteristics.
32	Output device	Transferring data to the I/O device (output operation).
33	Input device	Transferring data from the I/O device (input operation).
34	Output device	Transferring data to the I/O device (output operation).

### Notes About Device Addresses 31, 32, and 34:

1. When device address 32 is specified, the data assigned to the shared variable is not transferred to the I/O device until the buffer is full or the output device is closed (assign 10 to the shared variable).
2. When device address 31 or 34 is specified, each time data is assigned to the shared variable, the data is immediately transferred to the output device.
3. You can transfer multiple records to the I/O device faster using device address 34 rather than device address 32.



#### For BASIC

Device Address	Description	Use
A08	Command device	Specifying the user device characteristics.
A02	Input device	Transferring data from the I/O device using a GET statement (input operation).
A04	Output device	Transferring data to the I/O device using a PUT or PRINT statement (output operation).
A20	Input device	Loading a BASIC program from the I/O device using a LOAD command (input operation).
A40	Output device	Listing or punching a BASIC program on the I/O device using a SAVE command (output operation).

**Note:** Device addresses A20 and A40 can also be used for input/output operations using PUT, PRINT, and GET statements. However, it is recommended that these device addresses only be used for loading, listing, or punching a BASIC program.

## Chapter 3. User-Specified Device Characteristics

The user-specified device characteristics specify control information about the I/O device to the 5100. These characteristics are specified in a character constant with a comma separating each characteristic.

*Note:* Blanks must not be used as separators in the character constant.

Some device characteristics have a default value that is used if you do not specify that device characteristic. The device characteristics can be specified in any order and at any time the command device is open. Following are examples of specifying several device characteristics using the APL and the BASIC languages:

The diagram illustrates two code snippets with annotations. The first snippet is APL code: `A←'I/82,R/9600,P/E'`. An arrow points from the text "In APL, the device characteristics must be assigned to a shared variable." to the assignment operator `←`. Another arrow points from the text "Device characteristics must be separated by commas." to the commas within the character constant. The second snippet is BASIC code: `PRINT FL1, 'I/82,R/9600,P/E'`. An arrow points from the text "In BASIC, the device characteristics must be specified using a PRINT statement." to the `PRINT` statement.

In APL, the device characteristics must be assigned to a shared variable.

```
A←'I/82,R/9600,P/E'
```

Device characteristics must be separated by commas.

```
PRINT FL1, 'I/82,R/9600,P/E'
```

In BASIC, the device characteristics must be specified using a PRINT statement.

See Chapters 4 and 5 for more examples of specifying device characteristics using the APL or BASIC languages, respectively.

## SET INPUT BUFFER SIZE

The 5100 input buffer size should be set to the size of the largest record to be received plus two. The two extra bytes allow the 5100 to place an end-of-record and end-of-buffer code after the received record.

SYNTAX I/n

where n is an integer greater than 2 specifying the input buffer size.

*Note:* If the input buffer is not set large enough, loss of data can occur.

	5-bit	6-bit	7-bit	8-bit
Default Values	204	204	204	204

## SET OUTPUT BUFFER SIZE

The 5100 output buffer size should be set to the size of the largest record to be transmitted.

The 5100 takes the following actions when an entire record cannot be placed in the output buffer:

- For APL output operations using device address 32, the remaining portion of the record is placed in the output buffer after the data already in the buffer is transmitted.
- For APL output operations using device address 34, an EXCEEDS MAXIMUM RECORD LENGTH error message is displayed and no data is transmitted.
- For BASIC output operations, the remaining portion of the record is placed in the output buffer after the data already in the buffer is transmitted.

*Notes:*

1. For APL output operations and BASIC SAVE operations, the Serial I/O Adapter feature always allocates at least 204 bytes for the output buffer.
2. When this device characteristic is specified and the output device is open, the output buffer size is not changed. In this case, the output device must be closed and then opened again before the output buffer size is changed.

SYNTAX O/n

where n is an integer greater than 2 specifying the output buffer size.

	5-bit	6-bit	7-bit	8-bit
Default Values	204	204	204	204

## SET DATA RATE

The data rate specified depends on the requirements of the attached device. The data rate can be set from 20 bps (bits per second) to:

2400.5 bits per second when using 5-bit code

9600.5 bits per second when using 6-, 7-, or 8-bit codes

Also, the data rate can be specified to a half cycle, for example, 134.5 bps.

SYNTAX  $R/n[.5]$

where n is a positive number specifying the data rate.

	5-bit	6-bit	7-bit	8-bit
Default Values	75	134.5	110	110

## SET CHARACTER OR NUMERIC MODE

(Only for BASIC input operations using a GET statement.)

When performing input operations with character-only data, you can specify that the data is to be automatically placed in the buffer in 18 character blocks with each block enclosed in single quotes and separated by a comma. This allows any character-only data to be read with a GET statement by automatically providing the necessary delimiters (single quotes and commas).

SYNTAX F/  $\begin{Bmatrix} C \\ N \end{Bmatrix}$

where:

- C (character mode) specifies that the data is to be automatically blocked into 18-character blocks.
- N (numeric mode) specifies that data is not to be automatically blocked into 18-character blocks. In this case, the necessary delimiters must be provided as part of the input data.

	5-bit	6-bit	7-bit	8-bit
Default Values	N	N	N	C

↑ The F/N device characteristic cannot be specified for an 8-bit code.

*Note:* When using the character mode (F/C), be sure the input buffer size (see *Set Input Buffer Size* in this chapter), is set large enough to contain the additional quotes and commas. The buffer size can be determined by the following formula:

$$\text{SIZE} = \text{RECORD} + \text{QUOTES} + \text{COMMAS} + 2$$

where:

- RECORD is the size of the input record. (For 8-bit code, the size of the input record is 2 times the number of 8-bit codes received.)
- QUOTES is the number of single quotes required to separate the record into 18-character blocks. This value can be obtained by dividing RECORD by 9 (if the result contains a fraction or odd number, round up the result to the nearest even whole number).
- COMMAS is the number of commas required to separate each 18-character block. This value can be obtained by dividing QUOTES by 2, then subtracting 1 from the result.

## SET PROMPTING OR NONPROMPTING MODE

The prompting mode is specified when the input device requires an input prompting character before sending data to the 5100. When the prompting mode is specified, the 5100 automatically transmits the input prompting character (see *Set Input Prompting Character*) each time it is ready to receive data.

When prompting mode is specified, the 5100 transmits the input prompting character to the I/O device to indicate that the 5100 is ready to receive data. During the transmission of the input prompting character, the 5100 becomes ready to receive data, thus eliminating any transmit-to-receive transition time.

Following are two examples of using the  $\square$  (break or long space character) in prompting mode:

1. Assume the 5100 is attached to another system that must receive a break before transmitting its own input prompting character. In this case, the 5100 input prompting character is set to  $\square$ . When the 5100 requests input from the other system, the  $\square$  break is transmitted and the other system responds with its own input prompting character. Now, the 5100 can transmit data to the other system.
2. (APL only) Assume the 5100 is receiving data from an I/O device that terminates transmit mode when a *break* is received. In this example, you want to stop receiving data before the I/O device is ready to stop transmitting data. If you press CMD 0 (the 0 key on the right side of the keyboard), the 5100:
  - a. Stops receiving data from the I/O device
  - b. Closes the I/O device
  - c. Assigns an empty vector and 0 0 return code to the shared variable

Also, in this example, assume you are using an APL user-defined function that contains statements to transmit a break by setting the input prompting character to  $\square$ , opening the I/O device for input, and requesting input (transmits the input prompting character). After you have pressed CMD 0, the APL user-defined function can check the shared variable for an empty vector and 0 0 return code. When these conditions are met, the APL function should branch to the statements required to transmit the input prompting character.

Since the input prompting character is also the break character, the I/O device terminates transmit mode. The 5100 then waits for an end of transmission character from the I/O device or, if specified, a receive timeout (see *Set Receive Timeout* in this chapter). After one of these conditions has occurred, the 5100 can transmit data to the I/O device.



SYNTAX  $H/\left\{\begin{matrix} P \\ N \end{matrix}\right\}$

where:

- P (prompting mode) specifies that the 5100 will transmit an input prompting character
- N (nonprompting mode) specifies that the 5100 will not transmit an input prompting character

	5-bit	6-bit	7-bit	8-bit
Default Values	N	P	N	N

## SET INPUT PROMPTING CHARACTER

If the H/P (prompting mode) device characteristic is specified, the 5-, 6-, 7-, or 8-bit code that is equivalent to the specified 5100 character in the output translation table is used as the input prompting character.

SYNTAX C/p

where p is the specified 5100 character

	5-bit	6-bit	7-bit	8-bit
Default Values	⊕	⌘	↑	11

## SET MODEM, IGNORE, TERMINAL, WAIT, OR SET MODE

The 5100 must operate in one of five modes (modem, ignore, terminal, wait, or set) when using the Serial I/O Adapter feature. These modes determine the conventions used by the 5100 when:

- Making a request to send data to the I/O device (turning on request to send [RTS])
- Indicating it is ready to receive data from the I/O device (turning on clear to send [CTS])

Following is a description of the conventions used in each mode:

- In modem mode, the I/O device must first request to send data (turns request to send on), then the 5100 indicates when it is ready to receive the data (turns clear to send on). The 5100 turns clear to send off only after the 5100 code for end of buffer is placed in the buffer and the I/O device turns request to send off.
- In ignore mode, the 5100 does not check for a request to send before indicating whether or not it is ready to receive data. The 5100 turns clear to send on when it is ready to receive data and turns clear to send off when the 5100 code for end of buffer is placed in the buffer.
- In terminal mode, the 5100 requests to send data (turns request-to-send on), then waits for the I/O device to indicate when it is ready to receive the data (turns clear to send on). After the I/O device indicates it is ready to receive data, the 5100 begins transmitting.
- In wait mode, the 5100 requests to send data (turns request to send on), then waits for the I/O device to indicate when it is ready to receive data (turns clear to send on). After the I/O device indicates it is ready to receive data, the 5100 begins transmitting. (However, in this mode, four extra stop bits are added to allow the receiving device time to turn clear to send off.) As the 5100 is transmitting data, if the I/O device turns clear to send off, the 5100 stops transmitting and waits for the I/O device to turn clear to send on again.

The wait mode allows the I/O device to control the data transfer from the 5100. For example, assume you have a printer that turns clear to send on and receives data. However, when the data for one print line is received, the printer turns clear to send off and prints the data. After the data is printed, the printer turns clear to send on again and receives another line of data. In this case, the wait mode causes the 5100 to stop transmitting data and wait as the printer prints each line.

- In set mode, the 5100 sets the request-to-send or clear-to-send indication on and ignores any request-to-send or clear-to-send indications from the I/O device.

*Note:* Ignore, set, and wait modes are useful when the I/O device does not conform completely to the EIA RS232C /CCITT V.24-V.28 specification.

SYNTAX K/  $\left\{ \begin{array}{c} M \\ I \\ T \\ W \\ S \end{array} \right\}$

where:

- M specifies modem mode.
- I specifies ignore mode.
- T specifies terminal mode.
- W specifies wait mode.
- S specifies set mode.

	5-bit	6-bit	7-bit	8-bit
Default Values	M	M	M	M

## SET RECEIVE TIMEOUT

This user-specified device characteristic allows you to specify the *approximate* number of seconds (from 1 to 300) the 5100 will wait for the input device to transmit data. If the input device does not transmit data to the 5100 within the specified time, the 5100 code for end of record (hex E3) and end of buffer (hex FF) is placed in the buffer and the 5100 stops receiving data.

For APL operations, if there is no data in the input buffer and the input device does not transmit data to the 5100 within the specified time, the data returned to the shared variable is an empty vector. In this case, the return code assigned to the shared variable is 9 0.

For BASIC operations in numeric mode (see *Set Character or Numeric Mode* in this chapter), if there are not enough records in the buffer for the GET statement and the input device does not transmit data within the specified time, an ERROR 010 ddd is displayed and any existing data in the buffer is lost.

This user-specified device characteristic prevents the 5100 from waiting for an indefinite period of time when the input device does not have any more records to transmit and does not indicate when there is an end of file condition.

*Note:* In modem mode, after a receive timeout condition, the data is not returned to the APL or BASIC language until request to send from the I/O device turns off.

SYNTAX       $T / \left\{ \begin{array}{c} 0 \\ n \end{array} \right\}$

where:

- 0 specifies that the 5100 will wait for an indefinite period of time for the input device to transmit data (disable the receive timeout).
- n is a value from 1 to 300 that specifies the *approximate* number of seconds the 5100 will wait for data from the input device.

	5-bit	6-bit	7-bit	8-bit
Default Values	0	0	0	0

## SET INPUT NEW-LINE CHARACTER

The 5-, 6-, 7-, or 8-bit code that is equivalent to the specified 5100 character in the input translation table is used as the input new-line character. The input new-line character separates the records received by the 5100.

SYNTAX N/c

where c is the specified 5100 character.

	5-bit	6-bit	7-bit	8-bit
Default Values	C	Ψ	C	8D

## SET INPUT END-OF-BLOCK CHARACTER

The 5-, 6-, 7-, or 8-bit code that is equivalent to the specified 5100 character in the input translation table is used as the input end-of-block character.

SYNTAX E/c

where c is the specified 5100 character.

	5-bit	6-bit	7-bit	8-bit
Default Values	C	Ψ	C	8D

## SET OUTPUT NEW-LINE CHARACTER

The 5-, 6-, 7-, or 8-bit code that is equivalent to the specified 5100 character in the output translation table is used as the output new-line character.

SYNTAX L/c

where c is the specified 5100 character.

	5-bit	6-bit	7-bit	8-bit
Default Values	C	I	C	8D

## SET OUTPUT END-OF-BLOCK CHARACTER

The 5-, 6-, 7-, or 8-bit code that is equivalent to the specified 5100 character in the output translation table is used as the output end-of-block character.

SYNTAX B/c

where c is the specified 5100 character.

	5-bit	6-bit	7-bit	8-bit
Default Values	∅	I	∅	0A

## SET ENABLE OR DISABLE INPUT NEW-LINE AND INPUT END-OF-BLOCK CHARACTERS

When performing input operations, the 5100 normally places an end-of-record (hex E3) code in the buffer when an input new-line character is received, or places an end-of-buffer (hex FF) code in the buffer when an input end-of-block character is received. You can disable this action normally taken by the 5100 to allow the 5100 to receive multiple records from the input device. In this case, the 5100 recognizes the data as only one record and continued to receive data until only two positions remain unused in the buffer or a receive timeout occurs (see *Set Receive Timeout* in this chapter). The 5100 then places an end-of-record and end-of-buffer code in the last two positions of the buffer.

SYNTAX      Z/  $\left\{ \begin{array}{c} \text{E} \\ \text{D} \end{array} \right\}$

where:

- E specifies that the 5100 places an end-of-record code or end-of-buffer code in the buffer when an input new-line character or input end-of-block character, respectively, is received.
- D specifies that the 5100 does not place an end-of-record code or end-of-buffer code in the buffer when an input new-line character or input end-of-block character, respectively, is received.

	5-bit	6-bit	7-bit	8-bit
Default Values	E	E	E	E

## SET ENABLE/DISABLE OUTPUT NEW-LINE AND OUTPUT END-OF-BLOCK CHARACTERS

When performing output operations, the 5100 normally transmits an output new-line and output end-of-block character when the 5100 code for end-of-record (hex E3) is encountered in the buffer (and, for BASIC only, the physical end-of-buffer is reached). You can disable this action normally taken by the 5100 if you want to transmit multiple records to the output device.

*Note:* When the 5100 user-specified character equivalent of the output new-line or output end-of-block character is encountered in the buffer, the output new-line or output end-of-block character, respectively, is always transmitted.

SYNTAX      Y/  $\left\{ \begin{array}{c} \text{E} \\ \text{D} \end{array} \right\}$

where:

- E specifies that the output new-line and output end-of-block characters are transmitted (enabled).
- D specifies that the output new-line and output end-of-block characters are not transmitted (disabled).

	5-bit	6-bit	7-bit	8-bit
Default Values	E	E	E	E

## SET IMMEDIATE DELAY

This user-specified device characteristic causes the 5100 to immediately delay (stop processing all statements or commands), for a specific period of time (in 1/10 second intervals) whenever it is sent to the command device. For example, if you specify 3000, the 5100 delays for 300 seconds (five minutes) before processing any more statements or commands. This device characteristic can be specified any time the command device is open and can be specified more than once by an APL or BASIC program.

This device characteristic is usually used when the 5100 is attached to an input device that takes measurements at specific time intervals (for example, the room temperature).

SYNTAX D/n

where n is an integer from 1 to 65535 specifying the number of 1/10 seconds to delay.

## SET APL OR BASIC TRANSLATION TABLE

(Only for 6-Bit code.)

Appendix C shows the equivalent 6-bit code for each 5100 APL or BASIC character. During input operations, the 6-bit code is translated into the equivalent 5100 APL or BASIC character (depending on which translation table is used). During output operations, each 5100 character is translated into the equivalent 6-bit code (depending on which translation table is used).

This device characteristic specifies which 6-bit code translation table to use. For example, suppose you are using a BASIC program to transfer data to a device that prints APL characters, then you would want to specify the APL translation table.

SYNTAX A/{A  
B}

where:

- A specifies the APL translation table
- B specifies the BASIC translation table

	5-bit	6-bit	7-bit	8-bit
Default Values		A		



## CHANGE THE OUTPUT TRANSLATION TABLE

(Only for 5-, 6-, and 7-Bit codes.)

Each 5100 character is converted into the equivalent 5-, 6-, or 7-bit code (using an output translation table), before being transmitted to the I/O device. Appendixes B, C, and D show the equivalent 5-bit, 6-bit, and 7-bit codes for each 5100 character. The 5-, 6-, or 7-bit code equivalent of the 5100 character can be changed (by changing the output translation table) as follows:

**SYNTAX** `c>col/row`

where *c* is the 5100 character and *col/row* is the *col/row* value (see *Col/Row* column in Appendixes B, C, and D) of the new equivalent 5-, 6-, or 7-bit code.

An example of changing the APL 6-bit code output translation table:

To change the translation of the 5100 character T from the 6-bit code for the EBCD character *t* (*col/row* = 103) to the 6-bit code for the EBCD character *x* (*col/row* = 107), enter T>107. Now, both 5100 APL characters T and X will be converted into the 6-bit code for the EBCD character *x*.

An example of changing the 7-bit code output translation table:

To change the translation of the 5100 character P from the 7-bit code for the ASCII character *p* (*col/row* = 700) to the 7-bit code for the ASCII character *z* (*col/row* = 710), enter P>710. Now, both 5100 characters P and Z will be converted into the 7-bit code for the ASCII character *z*.

## CHANGE THE INPUT TRANSLATION TABLE

(Only for 5-, 6-, and 7-Bit codes.)

Each 5-, 6-, or 7-bit code received by the 5100 is converted into the equivalent 5100 character (using an input translation table). Appendixes B, C, and D show the equivalent 5100 character for each 5-, 6-, or 7-bit code. The 5100 character equivalent of the 5-, 6-, or 7-bit code can be changed (by changing the input translation table), as follows:

SYNTAX `c < col/row`

where *c* is the 5100 character and *col/row* is the *col/row* value (see the *Col/Row* column in Appendixes B, C, and D) of the 5-, 6-, or 7-bit code to be translated into the specified 5100 character.

An example of changing the 6-bit code BASIC input translation table:

To change the translation of the 6-bit code for the EBCD character *s* (*col/row* = 102) from the 5100 BASIC character *S* to the 5100 BASIC character *X*, enter *X* < 102. Now, the 6-bit codes for both EBCD characters *s* and *X* will be converted into the 5100 BASIC character *X*.

An example of changing the 7-bit code input translation table:

To change the translation of the 7-bit code for the ASCII character *p* (*col/row* = 700) from the 5100 character *P* to the 5100 character *Z*, enter *Z* < 700. Now, the 7-bit codes for both ASCII characters *p* and *z* will be converted into the 5100 character *Z*.

## SET INPUT/OUTPUT PARITY

(Only for 7-Bit code.)

The input/output parity can be set to odd parity or even parity depending upon the requirements of the I/O device, or parity checking can be turned off.

SYNTAX  $P/\left\{\begin{matrix} E \\ O \\ N \end{matrix}\right\}$

where:

- E specifies even parity.
- O specifies odd parity.
- N specifies no parity checking on input and a zero parity bit transmitted on output.

	5-bit	6-bit	7-bit	8-bit
Default Values		0	E	

The input/output parity cannot be changed when using 6-bit code. An error occurs if the device characteristic is specified.

## SET NUMBER OF STOP BITS

(Only for transmitting 7- and 8-Bit codes.)

The number of stop bits between characters can be set to 1 or 2. Normally, it doesn't matter whether the number of stop bits is one or two; however, some I/O devices require two stop bits between characters.

SYNTAX S/n

where n is the specified number of stop bits.

	5-bit	6-bit	7-bit	8-bit
Default Values	1.5	1	2	2

The number of stop bits cannot be changed when using 5- and 6-bit codes. An error occurs if the device characteristic is specified.

## USER-SPECIFIED DEVICE CHARACTERISTIC AND DEFAULT VALUE SUMMARY

User-Specified Device Characteristics	Syntax	Default Values			
		5-Bit	6-Bit	7-Bit	8-Bit
Set input buffer size	I/n	204	204	204	204
Set output buffer size	O/n	204	204	204	204
Set data rate	R/n	75	134.5	110	110
Set character or numeric mode (5, 6, and 7-bit code)	F/ $\begin{Bmatrix} N \\ C \end{Bmatrix}$	N	N	N	C <sup>1</sup>
Set prompting or nonprompting mode	H/ $\begin{Bmatrix} N \\ P \end{Bmatrix}$	N	P	N	N
Set modem, ignore, terminal, wait, or set mode	K/ $\begin{Bmatrix} M \\ I \\ T \\ W \\ S \end{Bmatrix}$	M	M	M	M
Set receive timeout	T/ $\begin{Bmatrix} 0 \\ n \end{Bmatrix}$	0	0	0	0
Set input prompting character	C/p	⊗	⌘	↑	11
Set input new-line character	N/c	␣	⌘	␣	8D
Set input end-of-buffer character	E/c	␣	⌘	␣	8D
Set output new-line character	L/c	␣	I	␣	8D
Set output end-of-buffer character	B/c	␣	I	␣	0A
Set enable or disable input new-line and input end-of-block characters	Z/ $\begin{Bmatrix} E \\ D \end{Bmatrix}$	E	E	E	E
Set enable or disable output new-line and output end-of-block characters	Y/ $\begin{Bmatrix} E \\ D \end{Bmatrix}$	E	E	E	E

<sup>1</sup> These default values cannot be changed and an error occurs if the device characteristic is specified.

User-Specified Device Characteristics	Syntax	5-Bit	Default Values		
			6-Bit	7-Bit	8-Bit
Set immediate delay	D/n				
Set APL or BASIC translation table (6-bit code only)	A/ $\begin{Bmatrix} A \\ B \end{Bmatrix}$		A		
Change the output translation table (5-, 6-, and 7-bit code)	c>col/row				
Change the input translation table (5-, 6-, and 7-bit code)	c<col/row				
Set input/output parity (7-bit code only)	P/ $\begin{Bmatrix} E \\ O \\ N \end{Bmatrix}$		0 <sup>1</sup>	E	
Set number of stop bits (7- and 8-bit code)	S/ $\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}$	1.5 <sup>1</sup>	1 <sup>1</sup>	2	2

---

<sup>1</sup> These default values cannot be changed and an error occurs if the device characteristic is specified.

## Chapter 4. Using the APL Language and the Serial I/O Adapter Feature

Once the 5-, 6-, 7-, or 8-bit code is selected and the Serial I/O Adapter program is loaded into the workspace, the user can:

- Load or copy a stored workspace, containing user-defined functions and variables, into the active workspace.

*Note:* Because the storage required for the Serial I/O Adapter program is subtracted from the active workspace, you might not be able to load a stored workspace that was written to tape using the )CONTINUE command.

- Perform input or output operations using an APL shared variable. Input or output operations that are going to be done more than once should be defined in a user-defined function.

### OUTPUT OPERATIONS

When you use device address 32, the data assigned to the shared variable is placed in the buffer until the buffer is full (see *Output End-of-Record and End-of-Buffer Conditions* in Chapter 2), or until the output device is closed. Then the data is transmitted until:

- The end of buffer is reached
- The ATTN key is pressed (Any remaining data in the buffer is lost.)
- An error occurs

*Note:* Records are placed into the buffer until the buffer is full; therefore, data might not be immediately transmitted to the output device. However, closing the device (assigning an empty vector to the shared variable) causes the data to be transmitted immediately.

When you use device address 34, the data assigned to the shared variable is immediately transmitted to the output device until:

- The transmission is complete
- The ATTN key is pressed (Any remaining data in the buffer is lost.)
- An error occurs

When you use device address 34, you can transmit multiple records faster because the output device does not have to be closed to transmit each record. Also, during all output operations, the data transmitted is a direct translation (character-by-character) of the data in the buffer.

## INPUT OPERATIONS

During input operations, data is received from the input device and placed into the buffer until:

- The buffer is full
- An input end-of-block character is received by the 5100
- The CMD key is held and the 0 key (on the right side of the keyboard) is pressed. (An empty vector is assigned to the shared variable and the input device is closed.)
- The ATTN key is pressed

*Note:* It is recommended that you use the CMD 0 key rather than the ATTN key to terminate an input operation.

- An error occurs

The data placed into the buffer is a direct translation of the data received. If the input new-line character and the input end-of-block character are the same character, the data is immediately assigned to the shared variable; the shared variable can then be used in an expression in the active workspace.

## End-of-File Conditions During APL Input Operations

During an input operation, an end-of-file condition means that the input device does not have any more records to transmit to the 5100. The Serial I/O Adapter feature recognizes an end-of-file condition when:

- The input device indicates end of file by dropping the data terminal ready signal. (See Chapter 4, *Connector and Pin Assignments* for more information about the data terminal ready pin.)
- The CMD key is held and the 0 key (on the right side of the keyboard) is pressed.

*Note:* CMD 0 is generally used when the input device does not indicate an end-of-file condition, even though it does not have any more records to transmit to the 5100. In this case, the 5100 will wait for the input device to transmit data until CMD 0 is pressed.

When the Serial I/O Adapter feature recognizes an end-of-file condition, an empty vector is automatically assigned to the shared variable and the input device is closed.

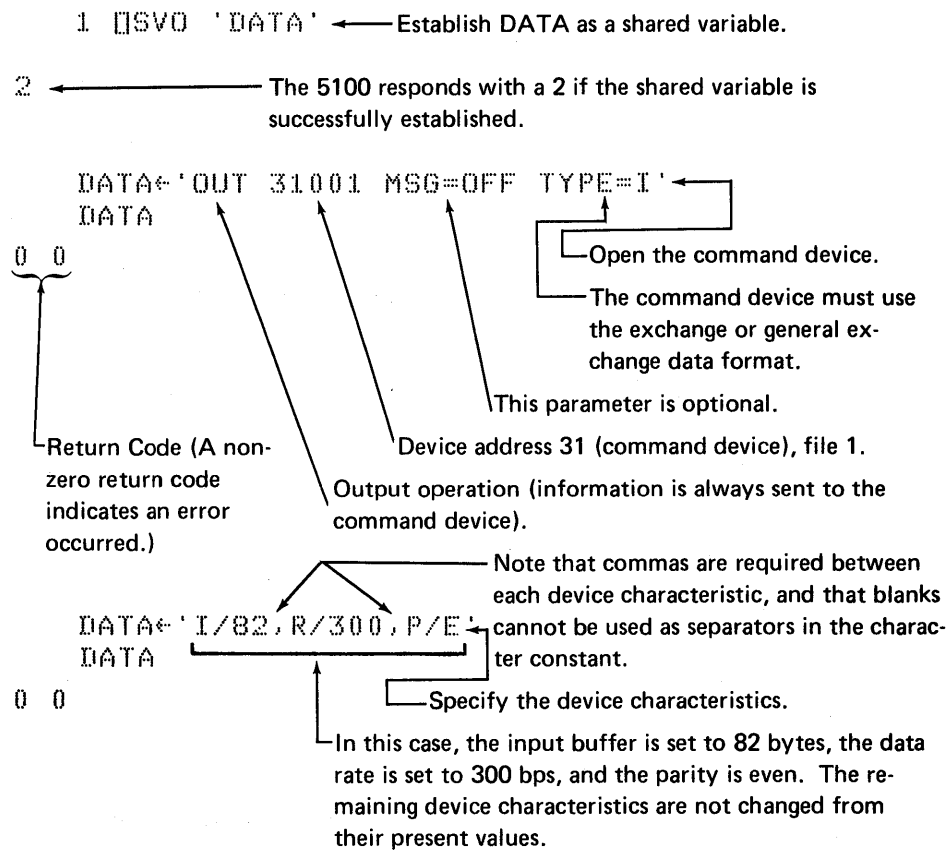
## ESTABLISHING A SHARED VARIABLE AND SPECIFYING THE DEVICE CHARACTERISTICS

Before input and/or output operations can be performed, a variable must be established as a shared variable and the device characteristics specified (see Chapter 3). To specify device characteristics, you must open the command device and specify the device characteristics. If the command device is not opened and the device characteristics specified, the default values for the device characteristics are used by the Serial I/O Adapter feature.

**Note:** Once the Serial I/O Adapter program is loaded and a value is specified for a device characteristic, that value remains in effect until:

- Another value is specified.
- The RESTART switch is pressed.
- The 5100 POWER ON/OFF switch is turned off.

Following is an example of establishing a shared variable and specifying some device characteristics:





```

VECTOR←'I/82,R/300,P/E'
DATA←VECTOR
DATA
0 0

```

← A character variable specifying the device characteristics can also be used.

The device characteristics can be specified in any order. Also, because the buffer is always automatically set to 204 when specifying device characteristics, assigning more than 203 characters (including commas) to the shared variable causes an EXCEEDS MAXIMUM RECORD LENGTH error message to be displayed.

```

DATA←' '
DATA
0 0

```

← Close the command device (assign an empty vector to the shared variable). The command device does not have to be closed. However, unless the command device is closed, another shared variable must be used to open the I/O device and transfer data (opening the I/O device and transferring data are discussed later in this chapter).

*Note:* If the command device is closed, it must be reopened before any other device characteristics can be specified.

## OPENING THE I/O DEVICE

After the shared variable is established and the device characteristics specified, the I/O device must be opened before data can be transferred. For example:

```

DATA←'IN 33001 MSG=OFF'
DATA
0 0

```

← Open an input device.  
 ← This parameter is optional when opening an input or output device.  
 ← Device address 33 (input device), file 1.  
 ← Input operation.

OR

```

DATA←'OUT 32001 TYPE=I'
DATA
0 0

```

← Open an output device using device address 32.  
 ← An output device must use the exchange or general exchange data format.  
 ← Device address 32 (output device), file 1.  
 ← Output operation.

OR

```

DATA←'OUT 34001'
DATA
0 0

```

← Open an output device using device address 34. In this case, you do not have to specify TYPE=1. However, only character scalars or vectors can be assigned to the shared variable.  
 ← Device address 34, file 1.  
 ← Output operation.

## TRANSFERRING DATA

After the input or output device is open, data can be transferred using the shared variable.

### Transmitting Data to the I/O Device (Output Operation)

#### Using Device Address 32:

When data (character scalar or vector only) is assigned to the shared variable, the data is placed into the buffer. If the data is to be transmitted immediately (before the buffer is full), the output device must be closed after each assignment to the shared variable. Once the device is closed, it must be reopened before more data can be assigned to the shared variable.

#### Using Device Address 34:

When data (character scalar or vector only) is assigned to the shared variable, the data is immediately transmitted to the I/O device.

*Note:* If the amount of data assigned to the shared variable exceeds the buffer size, the message **EXCEEDS MAXIMUM RECORD LENGTH** is displayed and no data is transmitted.

### Receiving Data From the I/O Device (Input Operation)

Data is received from the input device and placed into the buffer. Then, each time the shared variable is used, a new record from the buffer is assigned to it.

#### *Notes:*

1. If the input new-line and input end-of-block characters are the same character, only one record at a time is received from the input device and placed into the buffer.
2. When data is received from an input device, an empty vector is assigned to the shared variable if the only data in the 5100 input buffer is hex E3FF. In this case, the return code assigned to the shared variable is 9 0. The condition where only hex E3FF is in the buffer might be caused by the 5100 receiving only an input new-line and input end-of-block character (see *Input End-of-Record and End-of-Buffer Conditions* in Chapter 2) or by a receive timeout (see *Set Receive Timeout* in Chapter 3).

## CLOSING THE I/O DEVICE AND RETRACTING THE SHARED VARIABLE

After all the data is transferred, the I/O device should be closed and the shared variable name retracted. For example:

```
DATA←⍬ ← Close the I/O device by assigning an empty vector
          to the shared variable.
DATA
0 0
⍬SVR 'DATA' ← Retract the variable name being shared.
2
```

## 5100 CHARACTERS THAT REQUIRE USING THE ATOMIC VECTOR

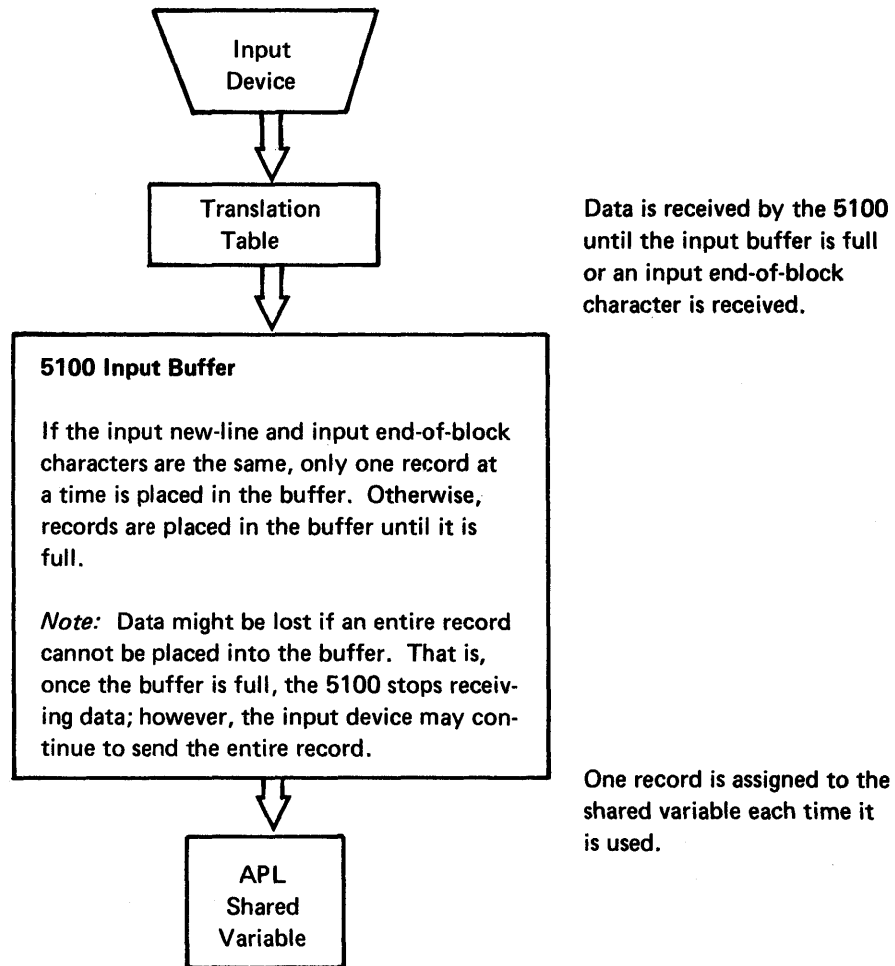
There are two 5100 characters (% and ¢) used with the Serial I/O Adapter feature that are not on the 5100 keyboard. To generate the % and ¢ characters, you must use the atomic vector. The indexes (location) of the % and ¢ characters in the atomic vector (assuming ⍬IO←1) are ⍬AV[173] and ⍬AV[184], respectively.

The catenate (,) function is used to place these characters into a record. For example:

```
SHAREDVARIABLE←'$1000 @ 4',⍬AV[173]
SHAREDVARIABLE
$1000 @ 4%
```

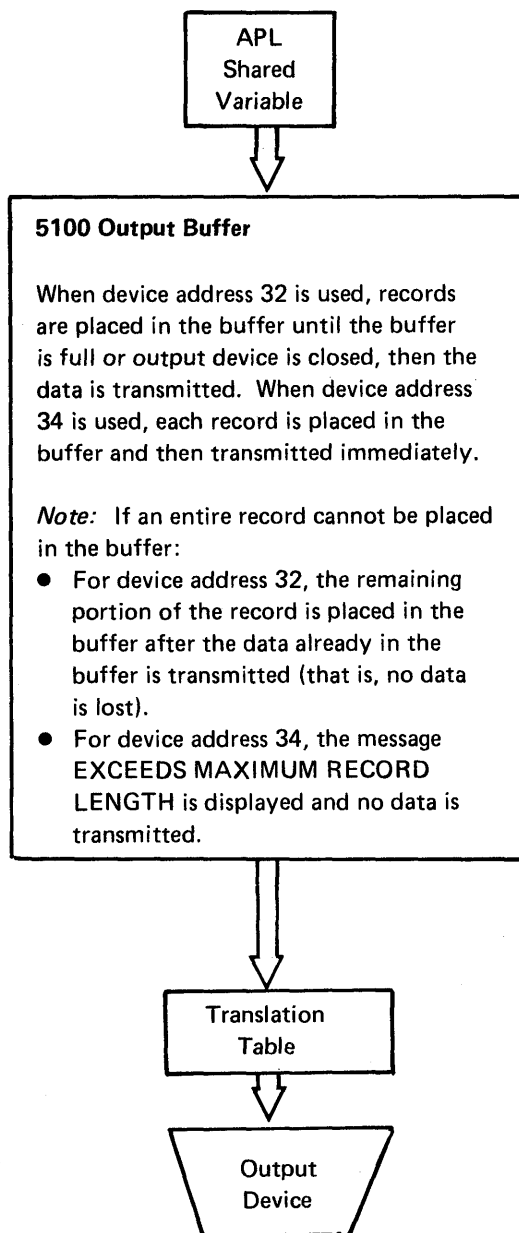
## INPUT DATA FLOW WHEN USING APL

The following diagram shows the data flow from an input device to the 5100:



## OUTPUT DATA FLOW WHEN USING APL

The following diagram shows the data flow from the 5100 to an output device:



A record is placed in the buffer when it is assigned to the shared variable.

## SAMPLE APL PROGRAM

The sample APL program shows how input and output operations can be performed using the IBM 5100 Serial I/O Adapter feature and an IBM 2741 Communications Terminal that uses EBCD code. To understand this example, you should be familiar with the following 2741 characteristics:

- Normally attaches to a modem (data set). Therefore, the modem cable attaches the 2741 to the 5100.
- Uses a 6-bit code.
- Requires a data rate of 134.5 bps.
- Uses the APL character format.
- Is placed in transmit mode when the POWER ON/OFF switch is turned on.
- When transmitting data:
  - a. Sends a **(D)** character to indicate the start of the data. If the APL translation table is specified, the 5100 character ] is equivalent to the **(D)** character. If the BASIC translation table is specified, the 5100 character # is equivalent to the **(D)** character.
  - b. Sends a **(C)** (**⌘**) character to indicate the end of the data (end of transmission).
- Must receive a **(D)** character to identify the start of a message and a **(C)** character to indicate the end of data.

```

V INΔOUTC[]V
V INΔOUT;CMDS;TEXT;A
[1] A
[2] A
[3] A
[4] A←1 [SVO 2 4 ρ'CMDSTEXT'
[5] →((1=√/A#2)/ERROR
[6] CMDS←'OUT 31001'
[7] →((1↑CMDS)≠0)/ERROR
[8] CMDS←'R/134.5,I/132,B/τ,K/S'
[9] →((1↑CMDS)≠0)/ERROR
[10] A
[11] A
[12] A
[13] A
[14] TEXT←'IN 33001'
[15] →((1↑TEXT)≠0)/ERROR
[16] 'THE 5100 IS WAITING FOR INPUT FROM THE 2741'
[17] TEXT
[18] TEXT←\0
[19] →((1↑TEXT)≠0)/ERROR
[20] A
[21] A
[22] A
[23] 'THE 5100 IS DELAYING FOR 1 MINUTE'
[24] CMDS←'D/600'
[25] →((1↑CMDS)≠0)/ERROR
[26] CMDS←\0
[27] →((1↑CMDS)≠0)/ERROR
[28] A
[29] A
[30] A
[31] TEXT←'OUT 32001 TYPE=I'
[32] →((1↑TEXT)≠0)/ERROR
[33] 'ENTER THE MESSAGE TO THE 2741'
[34] TEXT←']',0
[35] →((1↑TEXT)≠0)/ERROR
[36] 'NOW THE MESSAGE IS SENT TO THE 2741'
[37] TEXT←\0
[38] →((1↑TEXT)≠0)/ERROR
[39] A
[40] A
[41] A
[42] →RETRACT
[43] ERROR:'AN ERROR OCCURRED, TRY AGAIN'
[44] RETRACT:A←[SVR 2 4 ρ'CMDSTEXT'
V

```

Establish a shared variable and specify the device characteristics.

Check to make sure the operations are successful.

The Serial I/O Adapter feature uses the default value for any device characteristic not specified.

Turn clear to send (CTS) on for the 5100 and ignore any request to send (RTS) from the 2741.

Input operation.

Display this message to indicate when the 5100 is waiting for input from the 2741.

Display the input from the 2741.

Specify an immediate delay.

The 5100 will stop processing for one minute when this device characteristic is sent to the command device.

Close the command device.

Output operation.

The ] is converted into the (D) character for the 2741. The (D) character must precede any data to the 2741.

Close the output device to send the message to the 2741.

*Note:* The output device does not have to be closed to send the message when device address 34 is used for the output operation.

Retract the shared variable.

## Using the Sample APL Program

Before the sample APL program can be used, the following steps must be performed:

1. Attach the 2741 to the 5100 with the modem cable.
2. Place the 5100 BASIC/APL switch in the APL position and turn the power on to the 5100 and the 2741.

*Note:* If the power is already turned on, but the BASIC language is active, place the 5100 BASIC/APL switch in the APL position and press RESTART.

3. Place the tape cartridge containing the Serial I/O Adapter program into the built-in tape unit and enter the command:

)MODE COM

Press EXECUTE.

4. When the option menu is displayed, enter:

6

The message ,CLEAR WS, is displayed when the Serial I/O Adapter program is loaded into the workspace.

5. Load the sample program into the workspace (either by loading it from a tape cartridge or entering it from the keyboard).

Now, you can use the sample APL program to perform input and output operations with the 2741. For example:

1. Enter IN OUT and press EXECUTE to start the program.
2. When the 5100 displays THE 5100 IS WAITING FOR INPUT FROM THE 2741, enter FROM THE 2741 on the 2741 keyboard. As you are entering the data on the 2741 keyboard, notice the ← in the upper right corner of the 5100 display screen. This arrow moves in a back-and-forth motion as the 5100 receives data from the 2741 (for more information on the *Input and Output Process Arrows*, see Chapter 2). After you enter the data, press the RETURN key on the 2741. The message you entered is then displayed by the 5100.
3. After the 5100 displays the message from the 2741, it displays THE 5100 IS DELAYING FOR 1 MINUTE. Now, the 5100 will not process any statements or commands for one minute.
4. After the one minute delay, the 5100 displays ENTER THE MESSAGE TO THE 2741. When this message is displayed, enter FROM THE 5100 and press EXECUTE. Notice the → in the upper right corner of the 5100 display screen after the 5100 displays NOW, THE MESSAGE IS SENT TO THE 2741. This arrow moves in a back-and-forth motion as the 5100 transmits data to the 2741. The message you entered is then printed on the 2741.

This completes the execution of the sample APL program.



## Chapter 5. Using the BASIC Language and the Serial I/O Adapter Feature

Once the 5-, 6-, 7-, or 8-bit code is selected and the Serial I/O Adapter program is loaded into the work area, the user can:

- Enter a BASIC program from the keyboard or load a BASIC program from tape.
- Perform input and output operations with an I/O device using a BASIC program.
- Load a BASIC program from the I/O device.
- List or punch a BASIC program on the I/O device.

### OUTPUT OPERATIONS

BASIC output operations are initiated by PUT or PRINT statements in the user's program or by a SAVE command. During output operations, data is transmitted until:

- The physical end of the buffer is reached.
- An end-of-buffer character (hex FF) is encountered.
- The ATTN key is pressed. (Any remaining data in the buffer is lost.)
- An error occurs.

The data transmitted is a direct translation (character-by-character) of the data in the buffer.

If the PUT statement is used, records are placed into the buffer until the buffer is full. Therefore, the record might not be immediately transmitted to the output device. However, if the PRINT statement is used and a semicolon is not specified as the last delimiter, a record is immediately transmitted to the output device after it is placed into the buffer.

## INPUT OPERATIONS

BASIC input operations are initiated by a GET statement in the user's program or by a LOAD command. During input operations, data is received until:

- The buffer is full.
- An input end-of-block character is received by the 5100.
- The ATTN key is pressed.

During input operations initiated by a GET statement:

- a. If the character mode (see *Set Character or Numeric Mode* in Chapter 3) is specified, a record of 18 blanks is assigned to the referenced variables.
- b. If the character mode is not specified, an ERROR 010 A02 is returned.

During input operations initiated by a LOAD command, the entire work area is cleared, including all key groups.

- An error occurs.

Except when using the character mode (F/C), the data placed in the buffer is a direct translation (character-by-character) of the data received.

### End-of-File Conditions During BASIC Input Operations

During an input operation, an end-of-file condition means that the input device does not have any more records to transmit to the 5100. The Serial I/O Adapter feature recognizes an end-of-file condition when:

- The input device indicates an end-of-file by dropping the data terminal ready signal. (See Chapter 4, *Connector and Pin Assignments* for more information on the data terminal ready pin.)
- The CMD key is held and the 0 key (on the right side of the keyboard) is pressed.

*Note:* CMD 0 is generally used when the input device does not indicate an end-of-file condition, even though it does not have any more records to transmit to the 5100. In this case, the 5100 waits for the input device to transmit data until CMD 0 is pressed.

When the Serial I/O Adapter feature recognizes an end-of-file condition, an ERROR 010 A02 is returned.

*Note:* Because pressing CMD 0 causes the 5100 to recognize an end-of-file condition, CMD 0 should not be used as a function key when performing input operations.

## SPECIFYING THE DEVICE CHARACTERISTICS

Before input and/or output operations can be performed, the command device should be opened and a PRINT statement used to specify the device characteristics. If the command device is not opened and the device characteristics specified, the default values for the device characteristics are used by the Serial I/O Adapter feature.

**Note:** Once the Serial I/O Adapter program is loaded and a value is specified for a device characteristic, that value remains in effect until:

- Another value is specified
- The RESTART switch is pressed
- The 5100 POWER ON/OFF switch is turned off

Following is an example of specifying some device characteristics:

```
OPEN FL1, 'A08', OUT
```

↑ Open the command device.  
↑ Output operation (information is always sent to the command device).  
↑ Device address (command device).  
↑ Logical file reference.

```
PRINT FL1, 'I/82,R/300,P/E'
```

↑ Specify the device characteristics.  
↑ In this case, the input buffer is set to 82 bytes, the data rate is set to 300 bps, and the parity is even. The remaining device characteristics are not changed from their present values.

```
A$='I/82,R/300,P/E'
```

```
PRINT FL1,A$
```

↑ A character variable specifying the device characteristics can also be used.

The device characteristics can be specified in any order using one or more PRINT statements.

Although it is not required, it is recommended that the command device be closed after the device characteristics are specified. For example:

```
CLOSE FL1
```

← Now FL1 can be used as a file reference for some other purpose.

## OPENING THE I/O DEVICE FOR INPUT OR OUTPUT OPERATIONS

After the device characteristics are specified, the I/O device must be opened before an input or output operation can be performed. For example:

OPEN FL2, 'A04', OUT ← Open an output device.

↑  
↑  
↑  
Output operation.  
Device address A04 (for output operations using PUT or PRINT statements).  
Logical file reference.

or

OPEN FL3, 'A02', IN ← Open an input device.

↑  
↑  
↑  
Input operation.  
Device address A02 (for input operations using GET statements).  
Logical file reference.

### Note About Opening I/O Devices

If a BASIC program contains more than one OPEN statement with the same logical file reference (FL0 through FL9), an ERROR 152 occurs if a later input or output operation uses a larger buffer size than the buffer size first specified for that logical file reference. This error condition can be avoided by:

- Not using the same logical file reference for more than one OPEN statement in a program.
- Specifying the largest buffer size required for input or output operations (using the same logical file reference) first. For example:

```
0010 OPEN FL1, 'A08', OUT
0020 PRINT FL1, 'I/1000'
0030 CLOSE FL1
0040 OPEN FL2, 'A02', IN
0050 CLOSE FL2
```

Set the input buffer size (see Chapter 3) to specify the largest buffer size.

Now, the logical file reference FL2 can be used for input or output operations requiring buffer sizes up to 1000 bytes.

```
0090 OPEN FL1, 'A08', OUT
0100 PRINT FL1, 'O/512'
0110 CLOSE FL1
0120 OPEN FL2, 'A04', OUT
0130 PRINT FL2, 'OUTPUT DATA'
```

An output operation using the logical file reference FL2 and a 512-byte output buffer.

## PERFORMING INPUT OR OUTPUT OPERATIONS

After the I/O device is open, input operations can be performed using GET statements, or output operations can be performed using PUT or PRINT statements. For example:

```
GET FL2,A$,B$,C$ ← Input  
  
PUT FL3,X,27.23,'ABCD'  
PRINT FL3,X,27.23,'ABCD' ] ← Output
```

### Notes:

1. The PUT statement is not normally used except to transmit data to an offline storage media (for example, punched tape).
2. The PRINT statement has an advantage over the PUT statement because the data is left unblocked (only one record is placed in the buffer), and no delimiters (commas or quotes) are inserted.
3. The statement PRINT FL3,'ABC' transmits ABC plus as many blanks as in the remainder of the buffer. However, the following statements only transmit ABC plus an output new-line and output end-of-block character:  

```
A$=X'E3FF' ← Hex E3 is the 5100 end-of-record character and  
PRINT FL3,'ABC';A$ Hex FF is the 5100 end-of-buffer character.
```
4. To transmit a record containing only blanks, hex E3FF must follow the blanks in the buffer, for example:

```
A$=X'E3FF'  
PRINT FL3,' ';A$ ← Transmits a record of one blank  
character.
```

5. Character mode (see *Set Character* or *Regular Mode* in Chapter 3) allows any character-only input data to be read with a GET statement by automatically providing the necessary delimiters and placing the input data in the buffer in blocks of 18 characters.

## CLOSING THE INPUT OR OUTPUT DEVICE

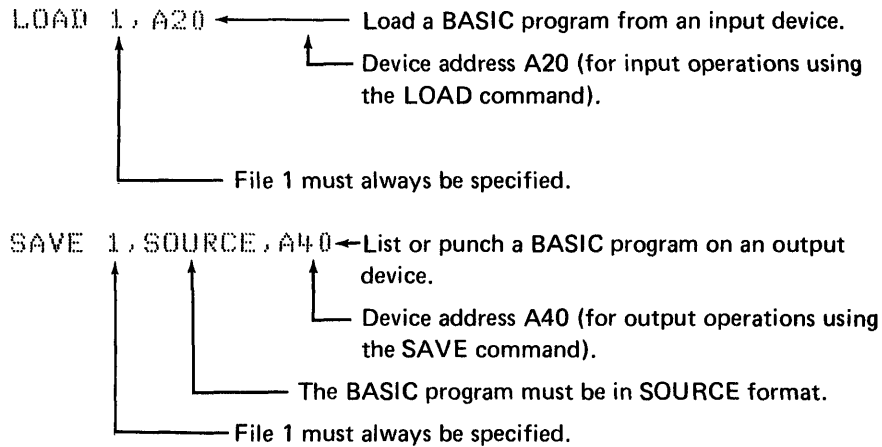
After the input or output operation is completed, the I/O device should be closed. For example:

```
CLOSE FL2  
  
CLOSE FL3
```

**Note:** If a device is left open, it is automatically closed when the program ends.

## LOADING, LISTING, OR PUNCHING A BASIC PROGRAM

After the device characteristics are specified (see Chapter 3), a BASIC program can be loaded (input), listed (output), or punched (output) on an appropriate I/O device using the LOAD or SAVE commands. For example:




### Notes:

1. When loading a BASIC program from an input device that does not indicate end-of-file conditions (see *End-of-File Conditions During BASIC Input Operations* in this chapter), CMD 0 must be pressed after the last record is transmitted to complete the operation. If the ATTN key is pressed instead of CMD 0, the entire work area is cleared.
2. When using 8-bit code, BASIC programs cannot be loaded, listed, or punched using an I/O device.



## SPECIAL CHARACTERS

### For a BASIC-Only Keyboard

Some of the special characters that are used with the Serial I/O Adapter feature are not shown on the BASIC-only keyboard. Figure 1 shows the combined BASIC/APL keyboard with the special characters shown on the keys. To enter the special character desired, press the corresponding key on the BASIC-only keyboard (Figure 2). For example, to enter the character  $\nabla$ :

1. Press and hold the shift key
2. Press the  key to enter the character  $\nabla$

Some of the special characters are formed using two characters. For example, to enter the character  $\Psi$ :

1. Press and hold the shift key
2. Press the  key to enter the character  $\nabla$
3. Press the backspace key
4. Press the  key to enter the character  $|$  over the character  $\nabla$ ; this forms the character  $\Psi$

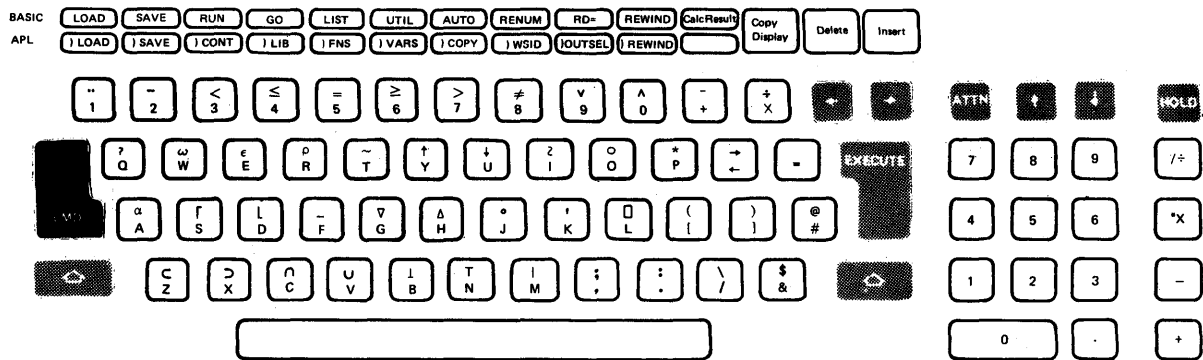


Figure 1. Combined BASIC/APL 5100 Keyboard

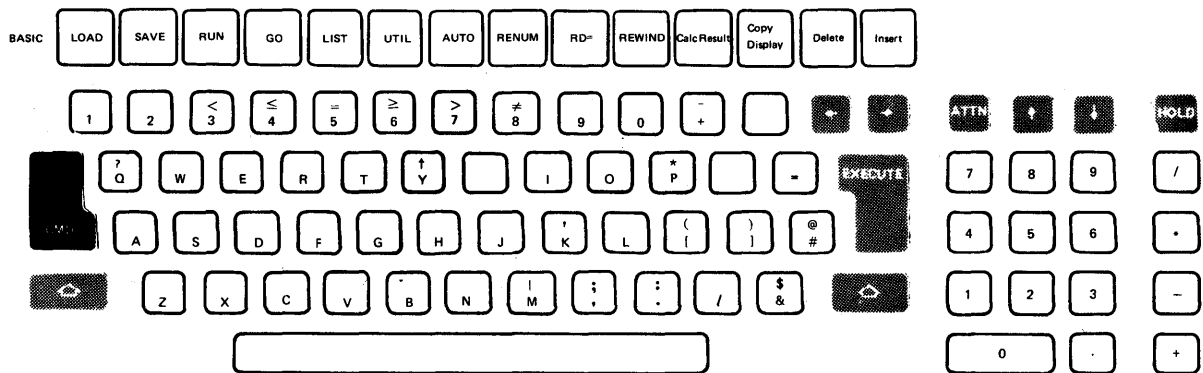


Figure 2. BASIC-Only 5100 Keyboard



The following chart shows the special characters and the BASIC-only key(s) that you must press to enter each character. Press and hold the shift key as you enter the character(s) unless indicated otherwise:

Special Character	Key(s)	Special Character	Key(s)	Special Character	Key(s)
$\alpha$	A	$\epsilon$	Z	$\tau$	I
$\Psi$	N J	$\Delta$	H M	$\rho$	R
T	N	$\Psi$	G M	L	D
$\perp$	B	$\nabla$	G	$\dots$	1
$\perp$	B J	$\backslash$	/ Lower Shift	$\wedge$	0
$\phi$	O M	$\times$	(see note)	$\sim$	O T
$\theta$	O +	$\div$		-	F
$\otimes$	O * P	-	2	$\Delta$	H
$\oslash$	O /	A	C J	$\sim \Delta$	H T
$\supset$	X	$\circ$	J	$\sim$	T
$\cap$	C	O	O	$\omega$	W
U	V	$\square$	L	<u>A through Z</u> Example:	
V	9	$\in$	E	<u>A</u>	A F
$\Gamma$	S	$\nabla$	L ' K		Lower Shift

**Note:** The blank key is located in the upper right corner of the alphameric keys.

Special Character	Key(s)
!	<div> <div>'</div> <div>K</div> <div>:</div> <div>.</div> <div>Lower shift</div> </div>
␣	<div> <div>L</div> <div></div> <div>(see note)</div> </div>
√	<div> <div>9</div> <div>T</div> </div>
÷	<div> <div>-</div> <div>+</div> <div>/</div> <div>Lower shift</div> </div>
λ	<div> <div>-</div> <div>+</div> <div>/</div> </div>
I	<div> <div>B</div> <div>N</div> </div>

**Note:** The blank key is located in the upper right corner of the alphameric keys.

### 5100 Characters That Are Not Shown On The Keyboard

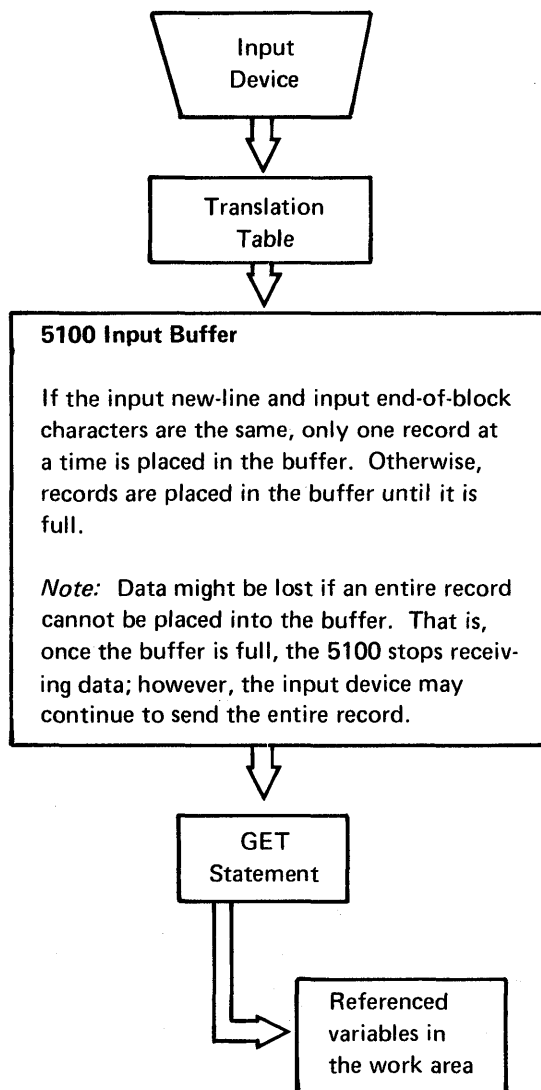
There are three 5100 characters ( — , % , and ¢ ) used with the Serial I/O Adapter feature that are not on the 5100 keyboard. To enter the — , % , or ¢ characters, you must use the hexadecimal values X'6D' , X'73' , or X'7D' , respectively. For example, the statements:

```
B$=X'73'
PRINT FL1, '90';B$
```

generates the record 90%.

## INPUT DATA FLOW WHEN USING BASIC

The following diagram shows the data flow from an input device to the 5100:



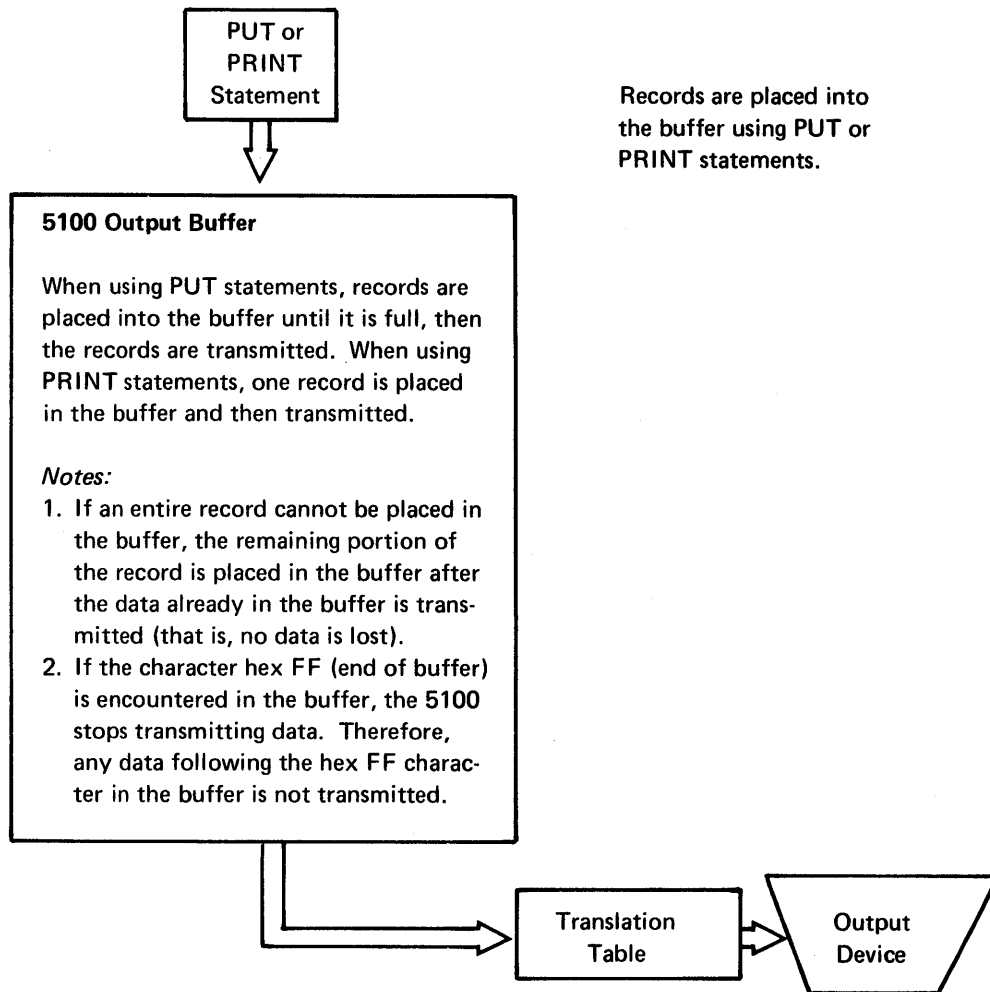
Data is received by the 5100 until the input buffer is full or an input end-of-block character is received.

Except when using the character mode, the data placed in the buffer is a direct translation of the data received.

The GET statement determines how many records at a time are removed from the buffer.

## OUTPUT DATA FLOW WHEN USING BASIC

The following diagram shows the data flow from the 5100 to an output device:



## SAMPLE BASIC PROGRAM

The sample BASIC program shows how input and output operations can be performed using the IBM 5100 Serial I/O Adapter feature and an IBM 2741 Communications Terminal that uses EBCD code. To understand this example, you should be familiar with the following 2741 characteristics:

- Normally attaches to a modem (data set). Therefore, the modem cable attaches the 2741 to the 5100.
- Uses 6-bit code.
- Requires a data rate of 134.5 bps.
- Uses the APL character format.
- Is placed in transmit mode when the POWER ON/OFF switch is turned on.
- When transmitting data:
  - a. Sends **(D)** character to indicate the start of the data. If the APL translation table is specified, the 5100 character ] is equivalent to the **(D)** character. If the BASIC translation table is specified, the 5100 character # is equivalent to the **(D)** character.
  - b. Sends a **(C)** (  $\Phi$  ) character to indicate the end of the data (end of transmission).
- Must receive a **(D)** character to identify the start of a message and a **(C)** character to indicate the end of data.

```

0010 REM
0020 REM
0030 REM
0040 OPEN FL1, 'A08', OUT
0050 PRINT FL1, 'R/134.5,O/132,I/132,B/7,K/S,F/C,A/B'
0060 REM
0070 REM
0080 REM
0090 REM
0100 REM
0110 REM
0120 OPEN FL2, 'A02', IN
0130 PRINT 'THE 5100 IS WAITING FOR INPUT FROM THE 2741'
0140 GET FL2,A#
0150 PRINT A#
0160 CLOSE FL2
0170 REM
0180 REM
0190 REM
0200 REM
0210 REM
0220 PRINT 'THE 5100 IS DELAYING 1 MINUTE'
0230 PRINT FL1, 'D/600'
0240 CLOSE FL1
0250 REM
0260 REM
0270 REM
0280 OPEN FL3, 'A04', OUT
0290 B#='X'E3FF'
0300 PRINT 'ENTER THE MESSAGE TO THE 2741 IN SINGLE QUOTES'
0310 INPUT C#
0320 F#='J'
0330 PRINT FL3,F#;C#;B#
0340 CLOSE FL3
0350 END

```

Specify the device characteristics.

Automatically block the input from the 2741 into 18-character blocks enclosed in single quotes.

Turn clear to send (CTS) on for the 5100 and ignore any request to send (RTS) from the 2741.

Input operation.

The Serial I/O Adapter feature uses the default value for any device characteristic not specified.

Display this message to indicate when the 5100 is waiting for input from the 2741.

Display the input from the 2741.

Specify an immediate delay (the command device is still open).

The 5100 stops processing for one minute.

Output operation.

The ] is converted into the Ⓚ for the 2741. The Ⓚ character must precede any data to the 2741.

Send the message to the 2741.

## Using the Sample BASIC Program

Before the sample BASIC program can be used, the following steps must be performed:

1. Attach the 2741 to the 5100 with the modem cable.
2. Place the 5100 BASIC/APL switch in the BASIC position and turn the power on to the 5100 and the 2741.

*Note:* If the 5100 power is already turned on, but the APL language is active, place the 5100 BASIC/APL switch in the BASIC position and press RESTART.

3. Place the tape cartridge containing the Serial I/O Adapter program into the built-in tape unit and enter the command:

UTIL MODE COM

Press EXECUTE.

4. When the option menu is displayed, enter:

6

The message ,READY, is displayed when the Serial I/O Adapter program is loaded into the work area.

5. Place the sample program into the work area (either by loading it from a tape cartridge or entering it from the keyboard).

Now, you can use the sample BASIC program to perform input and output operations with the 2741. For example:


1. Enter RUN and press EXECUTE to start the program.
2. When the 5100 displays THE 5100 IS WAITING FOR INPUT FROM THE 2741, enter FROM THE 2741 on the 2741 keyboard. As you are entering the data on the 2741 keyboard, notice the ← in the bottom right corner of the 5100 display screen. This arrow moves in a back-and-forth motion as the 5100 receives data from the 2741 (for more information on the *Input and Output Process Arrows*, see Chapter 2). After you enter the data on the 2741, press the RETURN key. The message you entered is then displayed by the 5100.
3. After the 5100 displays the message from the 2741, it displays THE 5100 IS DELAYING FOR 1 MINUTE. Now, the 5100 will not process any statements or commands for one minute.
4. After the one minute delay, the 5100 displays ENTER THE MESSAGE TO THE 2741 in single quotes. When this message is displayed, enter 'FROM THE 5100' and press EXECUTE. Notice the → in the bottom right corner of the 5100 display screen. This arrow moves in a back-and-forth motion as the 5100 transmits data to the 2741. The message you entered is then typed on the 2741.

This completes the execution of the sample BASIC program.



## Chapter 6. Serial I/O Adapter Characteristics and Specifications

The IBM 5100 Serial I/O Adapter feature has the following characteristics:

- Allows the 5100 to operate like a terminal or modem (under user-program control).
- Allows a data rate from 20 bps (bits per second) to 9600.5 bps; the data rate can be specified to a half cycle (for example, 134.5).
- Uses the 5-bit (no parity), 6-bit (plus parity), 7-bit (plus parity), and 8-bit (no parity) codes.
- The 5100 cannot transmit and receive simultaneously.
- When the 5100 is transmitting, it does not recognize a break (also called a long-space or receive interrupt).
- When 5-, 6-, or 7-bit code is being transmitted, a space is transmitted for a 3-character time span when a  character is processed. This 3-character time span is called a long space or break.
- Conforms to the EIA RS232C/CCITT V.24-V.28 standard for all signal levels and terminations.
- Deviates from the EIA RS232C/CCITT V.24-V.28 standard (for request to send [RTS] and clear to send [CTS] conventions) when the 5100 is in ignore, wait, or set mode.
- Uses all of the signals required by the EIA RS232C/CCITT V.24-V.28 interface for half duplex operation, when the 5100 is in modem or terminal mode.

*Note:* When the 5100 is being used in terminal mode, the direction of the signals at the 5100 serial I/O connector do not conform to the EIA RS232C/CCITT V.24-V.28 standard. However, the terminal cable interchanges certain pin assignments to provide EIA RS232C/CCITT V.24-V.28 standard pin assignments in terminal mode. See *Connector and Pin Assignments* for more information on the direction of the signals when using the 5100 in terminal mode.

- Recognizes an end-of-file condition when an I/O device drops the data terminal ready signal or if CMD 0 is entered from the 5100 keyboard during an input operation.

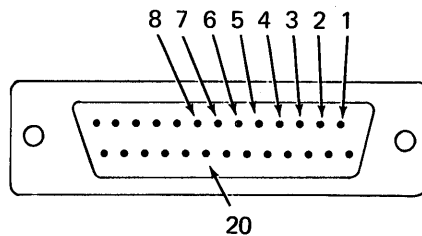
## CONNECTOR AND PIN ASSIGNMENTS

The 5100 connector socket is wired so that the 5100 simulates a modem (data set). When the 5100 is used as a terminal, the terminal cable interchanges the following signals at the I/O device connector socket:

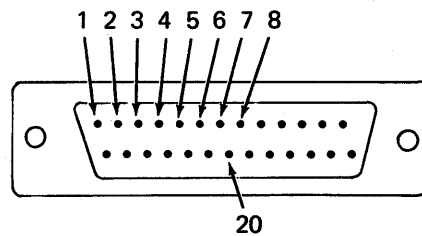
- Transmitted data with received data
- Request to send with clear to send
- Data terminal ready with data set ready

The pin assignments for the interface signals at the 5100 Serial I/O Adapter Connector socket, the modem cable I/O device connector socket, and the terminal cable I/O device connector socket are as follows:

**5100 Serial I/O Adapter Connector Socket and Modem Cable I/O Device Connector Socket**



**Terminal Cable I/O Device Connector Socket**



### Direction of Signal

Pin Number	At the 5100 Connector Socket	At the Modem Cable I/O Device Connector Socket	At the Terminal Cable I/O Device Connector Socket
1 Protective ground			
2 Transmitted data	To 5100	To 5100	From 5100
3 Received data	From 5100	From 5100	To 5100
4 Request to send	To 5100	To 5100	From 5100
5 Clear to send	From 5100	From 5100	To 5100
6 Data set ready	From 5100	From 5100	To 5100
7 Signal ground			
8 Received line signal detector	From 5100	From 5100	See note
20 Data terminal ready	To 5100	To 5100	From 5100

**Note:** The received line signal detector (pin 8) can be used as the originate signal, however, this signal is not supplied by the terminal cable.

## Appendix A. Set-Up Procedure

Use the following procedure to verify the operation of the Serial I/O Adapter feature:

1. Insert the tape cartridge containing the Serial I/O Adapter program into the 5100 built-in tape unit.
2. Enter the command to display the option menu:
  - a. If you are using APL, enter:  
    )MODE COM and press EXECUTE
  - b. If you are using BASIC, enter:  
    UTIL MODE COM and press EXECUTE
3. When the option menu is displayed, enter 8. This loads the Serial I/O Adapter program for 8-bit code.
  - a. If you are using APL, CLEAR WS is displayed when the Serial I/O Adapter program is loaded.
  - b. If you are using BASIC, READY is displayed when the Serial I/O Adapter program is loaded.
4. Enter the following program:
  - a. If you are using APL, enter:

```
          V SERIAL ;A;B
[1] B←1 [SVD 'A'
[2] A←'OUT 31001 TYPE=I'
[3] A←'D/600'
[4] A←\0
[5] 'TEST COMPLETE'
[6] V
```

} Press EXECUTE after each line.

- b. If you are using BASIC, enter:

```
0010 OPEN FL1, 'A08', OUT
0020 PRINT FL1, 'D/600'
0030 CLOSE FL1
0040 PRINT 'TEST COMPLETE'
0050 END
```

} Press EXECUTE after each line.

5. Execute the program:
  - a. If you are using APL, enter:  
SERIAL and press EXECUTE
  - b. If you are using BASIC, enter:  
RUN and press EXECUTE

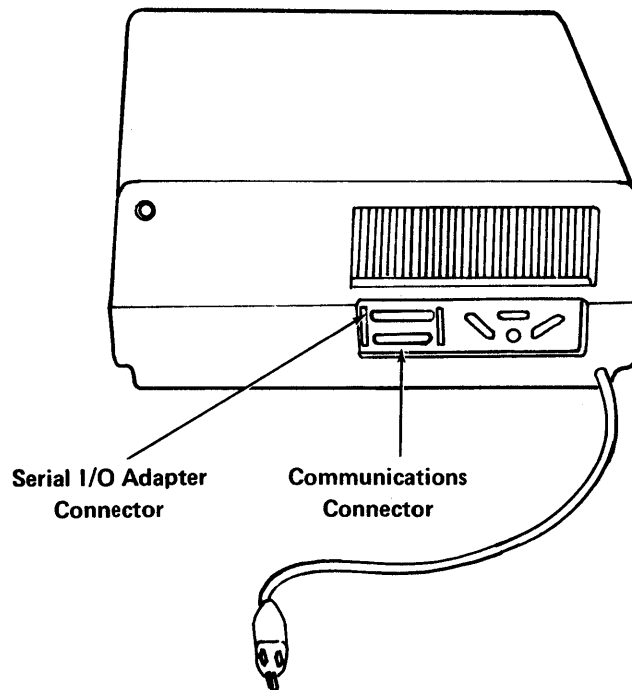
There should be one minute between the time you execute the program and the time the message TEST COMPLETE is displayed.

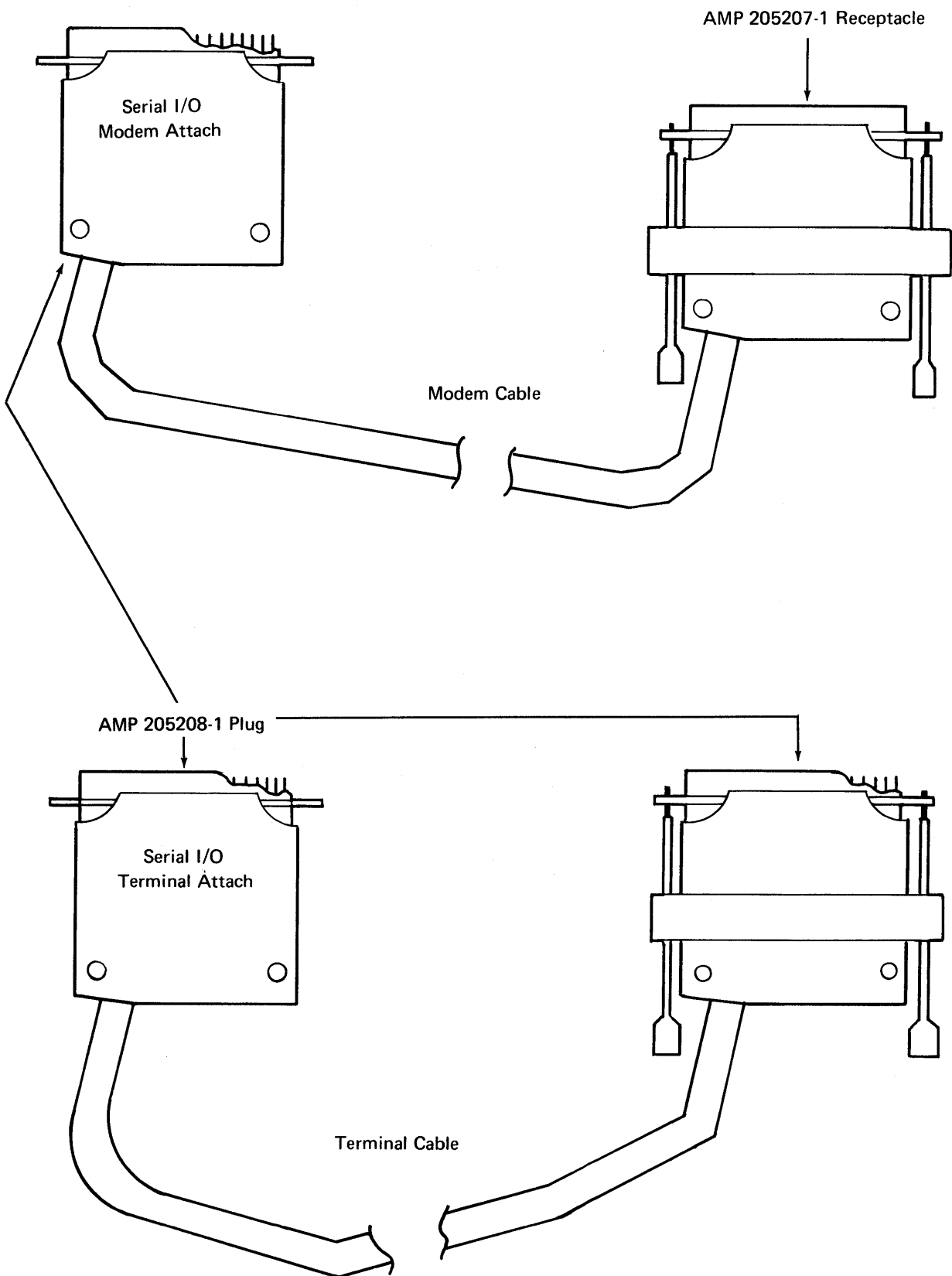
If the message TEST COMPLETE is displayed in one minute, the checkout of the Serial I/O Adapter feature is complete.

6. If the message TEST COMPLETE is not displayed in one minute, press RESTART and start from step 1 again. If you do not get the message TEST COMPLETE the second time, see the *Serial I/O Adapter Self Test* later in this appendix.

## ATTACHING THE EXTERNAL I/O DEVICE

There are two cables supplied with the Serial I/O Adapter feature. One cable is labeled Serial I/O Modem Attach (the modem cable) and the other cable is labeled Serial I/O Terminal Attach (the terminal cable). The modem cable, with an AMP 205207-1 receptacle, is used when the 5100 is used as data communications equipment. The terminal cable, with an AMP 205208-1 plug, is used when the 5100 is used as data terminal equipment.





## SERIAL I/O ADAPTER SELF TEST

The Serial I/O Adapter self test is provided to help you isolate problems between the 5100 and the attached device. To run the test:

1. Load the Serial I/O Adapter program.
2. Enter T when the option menu is displayed. This loads the test program and displays the following instructions:

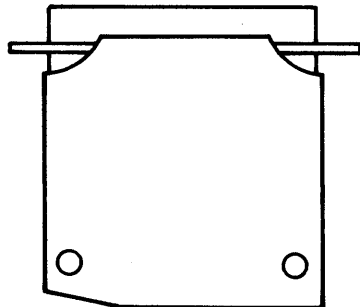
DISCONNECT THE SERIAL I/O  
ADAPTER CABLE FROM THE I/O  
DEVICE.

CONNECT THE WRAP CONNECTOR TO  
THE END OF THE SERIAL I/O  
CABLE.

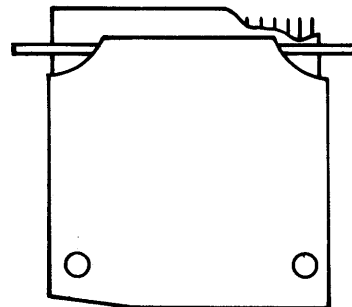
PRESS EXECUTE TO CONTINUE

-----

The two wrap connectors are stored in the *Maintenance Library Manual* binder. The wrap connector used for the Serial I/O Adapter connector or the modem cable is a plug (has pins), and the wrap connector used for the terminal cable is a receptacle (does not have pins), as shown in the following illustration:

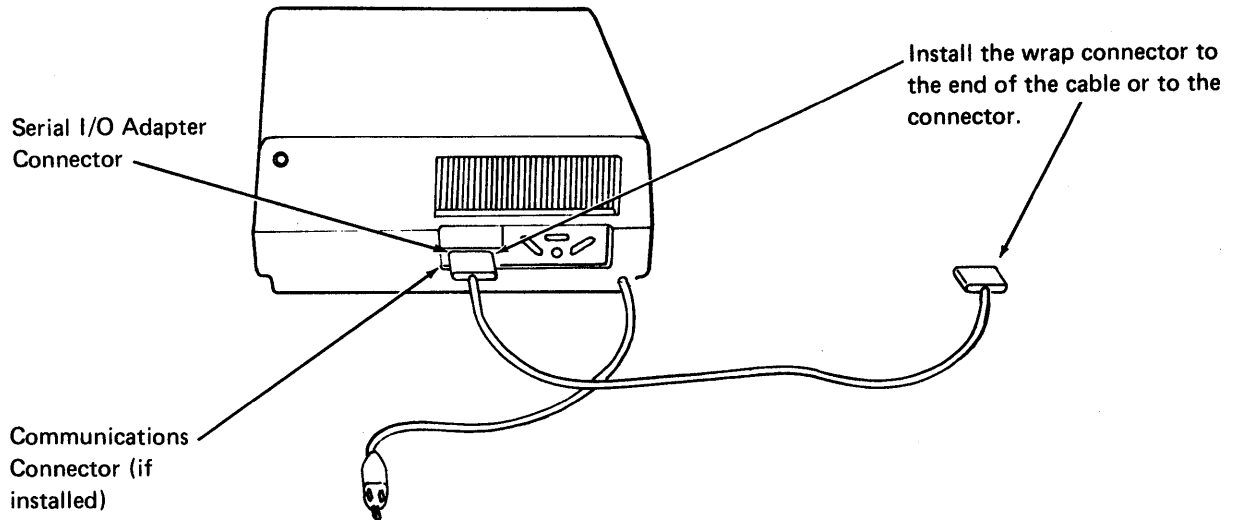


Wrap Connector for the  
Terminal Cable



Wrap Connector for the  
Serial I/O Adapter Connector  
or Modem Cable

3. Install the appropriate wrap connector to the end of the Serial I/O Adapter cable, or remove the Serial I/O Adapter cable at the back of the 5100 and install the appropriate wrap connector to the Serial I/O Adapter connector at the back of the 5100 as shown in the following illustration:



4. After you install the wrap connector, press the EXECUTE key. One of the following messages is displayed, giving you the results of the test:

NO TROUBLE FOUND

DISCONNECT THE WRAP CONNECTOR  
FROM THE END OF THE SERIAL I/O  
CABLE.

CONNECT THE SERIAL I/O CABLE TO  
THE I/O DEVICE.

PRESS EXECUTE TO RETURN TO OPTION MENU

-----

ERROR FOUND IN SERIAL I/O  
FEATURE.

CALL YOUR SERVICE  
REPRESENTATIVE.

DISCONNECT THE WRAP CONNECTOR  
FROM THE END OF THE SERIAL I/O  
CABLE.  
CONNECT THE SERIAL I/O CABLE TO  
THE I/O DEVICE.

PRESS EXECUTE TO RETURN TO OPTION MENU

-----

If the Serial I/O Adapter is functioning normally, the message NO TROUBLE FOUND is displayed. If the Serial I/O Adapter is not functioning normally, call your service representative.

5. To continue, press EXECUTE to return to the option menu.



## Appendix B. 5-Bit Code Conversion Chart

The 5-bit code conversion chart shows the equivalent 5100 character for each 5-bit code, and vice versa. This chart also shows the communications Baudot character for each 5-bit code. The *Col/Row* column in the conversion chart shows the two-digit value required to change the input or output translation tables. (See *Change the Output Translation Table* and *Change the Input Translation Table* in Chapter 3 for more information on changing the translation tables.)

**Note:** If the input or output translation table is changed by the user, the equivalent 5-bit code or 5100 character might not be the same in the translation table as shown in the conversion chart.

COL/ROW		5100 CHARACTERS		BIT VALUE					BAUDOT CHARACTERS	
LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	1	2	3	4	5	LETTERS (LOWER SHIFT)	FIGURES (UPPERSHIFT)
04	44	SPACE	SPACE			3			SPACE	SPACE
00	40	α	ω						BLANK	BLANK
10	50	n	n		2				LINE FEED	LINE FEED
02	42	c	c				4		CARRIAGE RETURN	CARRIAGE RETURN
33	73	⌘		1	2		4	5	FIGURES	
37	77	ψ		1	2	3	4	5	LETTERS	
30	70	A	-	1	2				A	-
23	63	B	?	1			4	5	B	?
16	56	C	:		2	3	4		C	:
22	62	D	\$	1			4		D	\$
20	60	E	3	1					E	3
26	66	F	!	1		3	4		F	!
13	53	G	&		2		4	5	G	&
05	45	H	#			3		5	H	#
14	54	I	8		2	3			I	8
32	72	J	'	1	2		4		J	'
36	76	K	(	1	2	3	4		K	(
11	51	L	)		2			5	L	)
07	47	M	.			3	4	5	M	.
06	46	N	,			3	4		N	,
03	43	O	9				4	5	O	9
15	55	P	0		2	3		5	P	0
35	75	Q	1	1	2	3		5	Q	1
12	52	R	4		2		4		R	4
24	64	S	@	1		3			S	BELL
01	41	T	5					5	T	5
34	74	U	7	1	2	3			U	7
17	57	V	;		2	3	4	5	V	;
31	71	W	2	1	2			5	W	2
27	67	X	/	1		3	4	5	X	/
25	65	Y	6	1		3		5	Y	6
21	61	Z	"	1				5	Z	"

## NOTES ABOUT LETTERS (LOWER SHIFT) AND FIGURES (UPPER SHIFT)

At the beginning of a transmit or receive operation, the 5100 assumes letters (lower shift). However, when all letters are being transmitted, the following characters can be used to change only the output device from letters (lower shift) to figures (upper shift), and vice versa:

- ⤴ — Sets the output device to figures (upper shift)
- ⤵ — Sets the output device to letters (lower shift)

Unless you specify otherwise, characters that can be both lower shift and upper shift (for example, `n`) are always transmitted to the I/O device as letters (lower shift). To transmit these characters as figures (upper shift), you must use the `c>col/row` user-specified device characteristic and specify the upper shift col/row value. For example, if the device characteristic `n>42` is specified, the `n` character is transmitted as figures (upper shift).

## Appendix C. 6-Bit Code Conversion Chart

The 6-bit code conversion chart shows the equivalent 5100 APL or BASIC character for each 6-bit code, and vice versa. This chart also shows the EBCD character for each 6-bit code. The *Col/Row* column in the conversion chart shows the three-digit value required to change the input or output translation tables. (See *Change the Output Translation Table* and *Change the Input Translation Table* in Chapter 3 for more information on changing the translation tables.)

**Note:** If the input or output translation table is changed by the user, the equivalent 6-bit code or 5100 character might not be the same in the translation table as shown in the conversion chart.

COL/ROW		APL CHARACTERS		BASIC CHARACTERS		BIT VALUES	EBCD CHARACTERS	
LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	B A 8 4 2 1 C	LOWER SHIFT	UPPER SHIFT
000	400	SPACE	SPACE	SPACE	SPACE		Space	Space
001	401	1	-	1	=	1	1	=
002	402	2	-	2	<	2	2	<
003	403	3	<	3	;	2 1 C	3	;
004	404	4	≤	4	:	4	4	:
005	405	5	=	5	%	4 1 C	5	%
006	406	6	≥	6	'	4 2 C	6	'
007	407	7	>	7	>	4 2 1	7	>
008	408	8	×	8	*	8	8	*
009	409	9	√	9	(	8 1 C	9	(
010	410	0	^	0	)	8 2 C	0	)
011	411	1	)	#	..	8 2 1	#	..
012	412	⋈	⋈	⋈	⋈	8 4 C	PN	PN
013	413	⋈	⋈	⋈	⋈	8 4 1	RS	RS
014	414	⋈	⋈	⋈	⋈	8 4 2	UC	UC
015	415	⋈	⋈	⋈	⋈	8 4 2 1 C	EOT	EOT
100	500	←	→	@	¢	A	@	¢
101	501	/	\	/	?	A 1 C	/	?
102	502	S	Γ	S	S	A 2 C	s	S
103	503	T	~	T	T	A 2 1	t	T
104	504	U	↓	U	U	A 4 C	u	U
105	505	V	u	V	V	A 4 1	v	V
106	506	W	ω	W	W	A 4 2	w	W
107	507	X	∞	X	X	A 4 2 1 C	x	X
108	508	Y	↑	Y	Y	A 8 C	y	Y
109	509	Z	c	Z	Z	A 8 1	z	Z
110	510	!	□	α	ω	A 8 2	,	
111	511	,	;	,	!	A 8 2 1 C	,	
112	512	⋈	⋈	⋈	⋈	A 8 4	BY	BY
113	513	⋈	⋈	⋈	⋈	A 8 4 1 C	LF	LF
114	514	⋈	⋈	⋈	⋈	A 8 4 2 C	EOB	EOB
115	515	⋈	⋈	⋈	⋈	A 8 4 2 1	PRE	PRE

COL/ROW		APL CHARACTERS		BASIC CHARACTERS		BIT VALUES	EBCD CHARACTERS	
LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	B A 8 4 2 1 C	LOWER SHIFT	UPPER SHIFT
200	600	+	-	-	-	B	-	-
201	601	J	°	J	J	B	j	J
202	602	K	'	K	K	B	k	K
203	603	L	□	L	L	B	l	L
204	604	M	I	M	M	B	m	M
205	605	N	T	N	N	B	n	N
206	606	O	o	O	O	B	o	O
207	607	P	*	P	P	B	p	P
208	608	Q	?	Q	Q	B	q	Q
209	609	R	p	R	R	B	r	R
210	610	#	@	!	!	B		
211	611	C	(	\$	!	B	\$	!
212	612	⌘	⌘	⌘	⌘	B	RES	RES
213	613	⌘	⌘	⌘	⌘	B	NL'	NL
214	614	⌘	⌘	⌘	⌘	B	BS	BS
215	615	⌘	⌘	⌘	⌘	B	IL	IL
300	700	x	÷	&	+	B A	&	+
301	701	A	α	A	A	B A	a	A
302	702	B	ι	B	B	B A	b	B
303	703	C	η	C	C	B A	c	C
304	704	D	L	D	D	B A	d	D
305	705	E	ε	E	E	B A	e	E
306	706	F	-	F	F	B A	f	F
307	707	G	∇	G	G	B A	g	G
308	708	H	Δ	H	H	B A 8	h	H
309	709	I	ι	I	I	B A 8	i	I
310	710	&	\$	Δ	∇	B A 8		
311	711	.	:	.	-	B A 8	.	-
312	712	⌘	⌘	⌘	⌘	B A 8 4	PF	PF
313	713	⌘	⌘	⌘	⌘	B A 8 4	HT	HT
314	714	⌘	⌘	⌘	⌘	B A 8 4 2	LC	LC
315	715	⌘	⌘	⌘	⌘	B A 8 4 2 1 C	DEL	DEL

## Definition of EBCD Control Characters

Control Character	Definition	Equivalent 5100 Character
PN	Punch on	~
PO	Punch off	~
BY	Bypass	~
RES	Restore	~
RS	Reader stop	⊘
LF	Line feed	⊘
NL	New line	I
HT	Horizontal tab	⊖
UC	Upper shift (output only)	⤴
LC	Lower shift (output only)	⤵
EOB	End of buffer	⊗
BS	Backspace	⤿
EOT	End of transmission ©	⌂
PRE	Prefix	⌂
IL	Idle (ignored on input operations)	⌂
DEL	Delete	⌂
EOA	End of address Ⓓ	# (for BASIC) ] (for APL)
	Long space (output only)	[:]

When the Serial I/O Adapter feature processes a [:] character, the transmitted data line is turned off for a three-character time span. The three-character time span is called a long space or break.

## NOTES ABOUT LOWER SHIFT AND UPPER SHIFT

The upper shift BASIC characters in the conversion chart are not always on the same keys as the corresponding lower shift BASIC characters. This chart simply indicates the equivalent upper shift 5100 BASIC character for each upper shift EBCD character.

At the beginning of a transmit or receive operation, the 5100 assumes lower shift. However, when all lower shift characters are being transmitted, the following characters can be used to change the output device from lower shift to upper shift, and vice versa:

- ⤴ — Sets the output device in upper shift
- ⤵ — Sets the output device in lower shift

Unless you specify otherwise, characters that can be both upper shift and lower shift (for example: ⌂) are always transmitted to the I/O device in lower shift. To transmit these characters in upper shift, you must use the c>col/row device characteristic and specify the upper shift col/row value. For example, if the device characteristic ⤵>415 is specified, the ⌂ character is transmitted in upper shift.

## Appendix D. 7-Bit Code Conversion Chart

The 7-bit code conversion chart shows the equivalent 5100 character for each 7-bit code, and vice versa. This chart also shows the ASCII character for each 7-bit code. The *Col/Row* column in the conversion chart shows the three-digit value required to change the input or output translation tables. (See *Change the Output Translation Table* and *Change the Input Translation Table* in Chapter 3 for more information on changing the translation tables.)

**Note:** If the input or output translation table is changed by the user, the equivalent 7-bit code or 5100 character might not be the same in the translation table as shown in the conversion chart.



COL/ROW	5100 CHARACTER	BIT VALUES							ASCII CHARACTER
		P	7	6	5	4	3	2	
000	α	0	0	0	0	0	0	0	NUL
001	τ	1	0	0	0	0	0	0	SOH
002	τ	1	0	0	0	0	0	1	STX
003	ι	0	0	0	0	0	0	1	ETX
004	ι	1	0	0	0	0	1	0	EOT
005	Φ	0	0	0	0	0	1	0	ENQ
006	Θ	0	0	0	0	0	1	1	ACK
007	Θ	1	0	0	0	0	1	1	BEL
008	ϙ	1	0	0	0	1	0	0	BS
009	ϙ	0	0	0	0	1	0	0	HT
010	η	0	0	0	0	1	0	1	LF
011	υ	1	0	0	0	1	0	1	VT
012	√	0	0	0	0	1	1	0	FF
013	ε	1	0	0	0	1	1	0	CR
014	Δ	1	0	0	0	1	1	1	SO
015	Ψ	0	0	0	0	1	1	1	SI
100	∇	1	0	0	1	0	0	0	DLE
101	↑	0	0	0	1	0	0	0	DC1
102	→	0	0	0	1	0	0	1	DC2
103	↓	1	0	0	1	0	0	1	DC3
104	←	0	0	0	1	0	1	0	DC4
105	x	1	0	0	1	0	1	0	NAK
106	÷	1	0	0	1	0	1	1	SYN
107	-	0	0	0	1	0	1	1	ETB
108	ι	0	0	0	1	1	0	0	CAN
109	η	1	0	0	1	1	0	0	EM
110	ο	1	0	0	1	1	0	1	SUB
111	ο	0	0	0	1	1	0	1	ESC
112	□	1	0	0	1	1	1	0	FS
113	ε	0	0	0	1	1	1	0	GS
114	ι	0	0	0	1	1	1	1	RS
115	ρ	1	0	0	1	1	1	1	US
200		1	0	1	0	0	0	0	SP
201	!	0	0	1	0	0	0	0	!
202	"	0	0	1	0	0	0	1	"
203	#	1	0	1	0	0	0	1	#
204	\$	0	0	1	0	0	1	0	\$
205	%	1	0	1	0	0	1	0	%
206	&	1	0	1	0	0	1	1	&
207	'	0	0	1	0	0	1	1	'
208	(	0	0	1	0	1	0	0	(
209	)	1	0	1	0	1	0	0	)
210	*	1	0	1	0	1	0	1	*
211	+	0	0	1	0	1	0	1	+
212	,	1	0	1	0	1	1	0	,
213	-	0	0	1	0	1	1	0	-
214	.	0	0	1	0	1	1	1	.
215	/	1	0	1	0	1	1	1	/
300	0	0	0	1	1	0	0	0	0
301	1	1	0	1	1	0	0	0	1
302	2	1	0	1	1	0	0	1	2
303	3	0	0	1	1	0	0	1	3
304	4	1	0	1	1	0	1	0	4
305	5	0	0	1	1	0	1	0	5
306	6	0	0	1	1	0	1	1	6
307	7	1	0	1	1	0	1	1	7
308	8	1	0	1	1	1	0	0	8
309	9	0	0	1	1	1	0	0	9
310	:	0	0	1	1	1	0	1	:
311	;	1	0	1	1	1	0	1	;
312	<	0	0	1	1	1	1	0	<
313	=	1	0	1	1	1	1	0	=
314	>	1	0	1	1	1	1	1	>
315	?	0	0	1	1	1	1	1	?

COL/ROW	5100 CHARACTER	BIT VALUES							ASCII CHARACTER
		P	7	6	5	4	3	2	
400	@	1	1	0	0	0	0	0	@
401	A	0	1	0	0	0	0	0	A
402	B	0	1	0	0	0	0	1	B
403	C	1	1	0	0	0	0	1	C
404	D	0	1	0	0	0	1	0	D
405	E	1	1	0	0	0	1	0	E
406	F	1	1	0	0	0	1	1	F
407	G	0	1	0	0	0	1	1	G
408	H	0	1	0	0	1	0	0	H
409	I	1	1	0	0	1	0	0	I
410	J	1	1	0	0	1	0	1	J
411	K	0	1	0	0	1	0	1	K
412	L	1	1	0	0	1	1	0	L
413	M	0	1	0	0	1	1	0	M
414	N	0	1	0	0	1	1	1	N
415	O	1	1	0	0	1	1	1	O
500	P	0	1	0	1	0	0	0	P
501	Q	1	1	0	1	0	0	0	Q
502	R	1	1	0	1	0	0	1	R
503	S	0	1	0	1	0	0	1	S
504	T	1	1	0	1	0	1	0	T
505	U	0	1	0	1	0	1	0	U
506	V	0	1	0	1	0	1	1	V
507	W	1	1	0	1	0	1	1	W
508	X	1	1	0	1	1	0	0	X
509	Y	0	1	0	1	1	0	0	Y
510	Z	0	1	0	1	1	0	1	Z
511	[	1	1	0	1	1	0	1	[
512	\	0	1	0	1	1	1	0	\
513	]	1	1	0	1	1	1	0	]
514	^	1	1	0	1	1	1	1	↑ or ^
515	_	0	1	0	1	1	1	1	_
600	A	0	1	1	0	0	0	0	\
601	a	1	1	1	0	0	0	0	a
602	B	1	1	1	0	0	0	1	b
603	C	0	1	1	0	0	0	1	c
604	D	1	1	1	0	0	1	0	d
605	E	0	1	1	0	0	1	0	e
606	F	0	1	1	0	0	1	1	f
607	G	1	1	1	0	0	1	1	g
608	H	1	1	1	0	1	0	0	h
609	I	0	1	1	0	1	0	0	i
610	J	0	1	1	0	1	0	1	j
611	K	1	1	1	0	1	0	1	k
612	L	0	1	1	0	1	1	0	l
613	M	1	1	1	0	1	1	0	m
614	N	1	1	1	0	1	1	1	n
615	O	0	1	1	0	1	1	1	o
700	P	1	1	1	1	0	0	0	p
701	Q	0	1	1	1	0	0	0	q
702	R	0	1	1	1	0	0	1	r
703	S	1	1	1	1	0	0	1	s
704	T	0	1	1	1	0	1	0	t
705	U	1	1	1	1	0	1	0	u
706	V	1	1	1	1	0	1	1	v
707	W	0	1	1	1	0	1	1	w
708	X	0	1	1	1	1	0	0	x
709	Y	1	1	1	1	1	0	0	y
710	Z	1	1	1	1	1	0	1	z
711	[	0	1	1	1	1	0	1	{
712	\	1	1	1	1	1	1	0	
713	]	0	1	1	1	1	1	0	}
714	^	0	1	1	1	1	1	1	~
715	_	1	1	1	1	1	1	1	DEL

The following chart can also be used to find the ASCII character for each 7-bit ASCII code:

BITS					b7 →	0	0	0	0	1	1	1	1
b4	b3	b2	b1	b5 →	b6 →	0	0	1	1	0	0	1	1
					Col	0	1	2	3	4	5	6	7
Row						0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	0	@	P	\	P
0	0	0	1	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	0	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	0	0	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	1	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	0	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	1	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	0	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	0	0	FF	FS	,	<	L	\	l	!
1	1	0	1	1	1	CR	GS	-	=	M	]	m	}
1	1	1	0	0	0	SO	RS	.	>	N	~	n	~
1	1	1	1	1	1	SI	US	/	?	O	-	o	DEL

## Appendix E. Device Address and User-Specified Device Characteristic Summary

### DEVICE ADDRESSES

#### For APL:

Device Address	Use	Syntax
31	Open the command device	SHAREDΔVARIABLE←'OUT 31001'
32	Open the output device	SHAREDΔVARIABLE←'OUT 32001 TYPE=I'
33	Open the input device	SHAREDΔVARIABLE←'IN 33001'
34	Open the output device	SHAREDΔVARIABLE←'OUT 34001'

#### For BASIC:

Device Address	Use	Syntax
A08	Open the command device	statement number OPEN FL1, 'A08', OUT
A02	Open the input device for a GET statement	statement number OPEN FL1, 'A02', IN
A04	Open the output device for a PUT or PRINT statement	statement number OPEN FL1, 'A04', OUT
A20	Open the input device and LOAD a BASIC program	LOAD 1, A20
A40	Open the output device and SAVE a BASIC program	SAVE 1, SOURCE, A40

## USER-SPECIFIED DEVICE CHARACTERISTICS

User-Specified Device Characteristics	Syntax	5-Bit	Default Values		
			6-Bit	7-Bit	8-Bit
Set input buffer size	I/n	204	204	204	204
Set output buffer size	O/n	204	204	204	204
Set data rate	R/n	75	134.5	110	110
Set character or numeric mode (5, 6, and 7-bit code)	F/ $\begin{Bmatrix} N \\ C \end{Bmatrix}$	N	N	N	C <sup>1</sup>
Set prompting or nonprompting mode	H/ $\begin{Bmatrix} N \\ P \end{Bmatrix}$	N	P	N	N
Set modem, ignore, terminal, wait, or set mode	K/ $\begin{Bmatrix} M \\ I \\ T \\ W \\ S \end{Bmatrix}$	M	M	M	M
Set receive timeout	T/ $\begin{Bmatrix} 0 \\ n \end{Bmatrix}$	0	0	0	0
Set input prompting character	C/p	⊕	⌘	↑	11
Set input new-line character	N/c	⌘	⌘	⌘	8D
Set input end-of-buffer character	E/c	⌘	⌘	⌘	8D
Set output new-line character	L/c	⌘	I	⌘	8D
Set output end-of-buffer character	B/c	⌘	I	⌘	0A
Set enable or disable input new-line and input end-of-block characters	Z/ $\begin{Bmatrix} E \\ D \end{Bmatrix}$	E	E	E	E
Set enable or disable output new-line and output end-of-block characters	Y/ $\begin{Bmatrix} E \\ D \end{Bmatrix}$	E	E	E	E

<sup>1</sup> These default values cannot be changed and an error occurs if the device characteristic is specified.

User-Specified Device Characteristics	Syntax	Default Values			
		5-Bit	6-Bit	7-Bit	8-Bit
Set immediate delay	D/n				
Set APL or BASIC translation table (6-bit code only)	A/ $\left\{ \begin{matrix} A \\ B \end{matrix} \right\}$		A		
Change the output translation table (5-, 6-, and 7-bit code)	c>col/row				
Change the input translation table (5-, 6-, and 7-bit code)	c< col/row				
Set input/output parity (7-bit code only)	P/ $\left\{ \begin{matrix} E \\ O \\ N \end{matrix} \right\}$		0 <sup>1</sup>	E	
Set number of stop bits (7- and 8-bit code)	S/ $\left\{ \begin{matrix} 1 \\ 2 \end{matrix} \right\}$	1.5 <sup>1</sup>	1 <sup>1</sup>	2	2

---

<sup>1</sup> These default values cannot be changed and an error occurs if the device characteristic is specified.

The error messages displayed when you are using the Serial I/O Adapter feature are associated with:

- The Serial I/O Adapter feature
- The APL or BASIC languages
- A nonzero return code assigned to an APL shared variable. (In this case, no error message is displayed when the MSG=OFF parameter is specified.)

Most of the error messages associated with the Serial I/O Adapter feature are displayed in the format:

ERROR eee ddd

where eee is the error code and ddd is the device address in error (031-034 for APL or A02-A08 for BASIC).

See the *IBM 5100 APL Reference Manual*, SA21-9213, or the *IBM 5100 BASIC Reference Manual*, SA21-9217, for a description of the error messages associated with the APL or BASIC languages.

*Note:* Some of the error messages displayed might also be followed by the message PRESS EXECUTE; for example:

ERROR 007 E80 PRESS EXECUTE

In this case, the only valid user response for the error message is to press EXECUTE to return to the APL or BASIC language.

See the *IBM 5100 APL Reference Manual*, SA21-9213, for a description of the error messages associated with a nonzero return code assigned to an APL shared variable.

The following list contains the error messages associated with the Serial I/O Adapter feature along with some possible causes for the error condition and a suggested user's response.

*Note:* For BASIC operations, any error deactivates the keys on the keyboard and causes the display screen to flash. Before you try to correct the error condition, press ATTN to stop the flashing display and then enter GO END to end the program.


Error Message	Cause	User's Response
ERROR 002 ddd	An invalid command was sent to an I/O device. For example, a REWIND (BASIC) or )REWIND (APL) command is issued to an I/O device.	You cannot specify the command to the I/O device.
	An invalid device characteristic was specified to the command device.	Respecify the device characteristics correcting the invalid device characteristic.
	The input buffer size was changed after the input device was opened.	Close the input device, change the buffer size, then reopen the input device.
	An invalid parameter was specified when opening the command device or an I/O device.	Reenter the statement correcting the invalid parameter.
ERROR 003 ddd	In modem mode, data terminal ready from the I/O device is off during a 5100 transmit operation. In terminal mode, data set ready from the I/O device is off during a 5100 transmit operation. These conditions can be caused by:	
	<ul style="list-style-type: none"> <li>• The I/O device does not conform to the EIA RS232C/CCITT V.24-V.28 standard interface.</li> </ul>	
	<ul style="list-style-type: none"> <li>• A defective cable.</li> </ul>	Replace the defective cable and retry the operation.
	<ul style="list-style-type: none"> <li>• The I/O device power is not on.</li> </ul>	Turn the I/O device power on and retry the operation.
	<ul style="list-style-type: none"> <li>• The I/O device is attached with the wrong cable.</li> </ul>	Use the correct cable and retry the operation (see <i>Appendix A. Set-Up Procedure</i> for more information on which cable to use).



Error Message	Cause	User's Response
ERROR 003 ddd (continued)	<p>In modem mode, request to send (RTS) from the I/O device dropped during a 5100 receive operation.</p> <p>In terminal mode, clear to send (CTS) from the I/O device dropped during a 5100 transmit operation.</p> <p>These conditions can be caused by:</p> <ul style="list-style-type: none"> <li>● The I/O device does not conform to the EIA RS232C/CCITT V.24-V.28 interface.</li> <li>● A defective cable.</li> <li>● The I/O device is attached with the wrong cable.</li> <li>● The I/O device power is not on.</li> <li>● The wrong mode (modem, ignore, terminal, or set) is specified.</li> <li>● Hardware malfunction.</li> </ul>	<p>Try using ignore or set mode (see Chapter 3).</p> <p>Replace the defective cable and retry the operation.</p> <p>Use the correct cable and retry the operation (see <i>Appendix A. Set-Up Procedure</i> for information on which cable to use).</p> <p>Turn the I/O device power on and retry the operation.</p> <p>Specify the correct mode (see Chapter 3) and retry the operation.</p> <p>Try the operation again. If the error occurs again, try to determine if the error is caused by the I/O device or the 5100. Then call the appropriate service representative.</p>

Error Message	Cause	User's Response
ERROR 004 ddd	Hardware malfunction.	Try the operation again. If the error occurs again, call the service representative.
ERROR 010 ddd (BASIC only)	<p>The 5100 recognized an end-of-file condition. This condition can be caused by:</p> <ul style="list-style-type: none"> <li>● Entering CMD 0.</li> <li>● In numeric mode, a receive timeout condition occurred and there were not enough records in the buffer for the GET statement. (In this case, any existing data in the buffer is lost.)</li> <li>● In modem mode, data terminal ready from the I/O device is off during a 5100 receive operation.</li> <li>● In terminal mode, data set ready from the I/O device is off during a 5100 receive operation.</li> </ul>	This is a warning message only.
ERROR 013 ddd	<p>The Serial I/O Adapter feature hardware is not installed or it is defective.</p> <p>The Serial I/O Adapter program is not loaded in user storage.</p>	Load the Serial I/O Adapter program.
ERROR 014 ddd	An invalid device address was specified.	Reenter the statement, correcting the invalid device address.

Error Message	Cause	User's Response
INVALID FILE PRESS EXECUTE	The Serial I/O Adapter feature cartridge is not in the built-in tape unit when an option is selected from the option menu or there is a problem with the Serial I/O Adapter feature cartridge.	Press EXECUTE to return to the APL or BASIC language. If the error is caused by the wrong tape cartridge in the built-in tape unit, insert the Serial I/O Adapter feature cartridge and load the Serial I/O Adapter program again. If the error is caused by the Serial I/O Adapter cartridge, call the service representative.
WORKSPACE FULL PRESS EXECUTE	The Serial I/O Adapter requires more storage than is available in your 5100.	Press EXECUTE to return to the APL or BASIC language.

**break:** 1. A 3-character time span transmitted by the 5100 when a  is encountered in the output buffer (for 5-, 6-, and 7-bit code only). 2. A request made by the 5100 for the input device to stop transmitting data.

**buffers:** A part of the 5100 user storage reserved for input and output operations.

**character mode:** (BASIC only) A mode of operation where character-only input data is automatically blocked into 18 characters with each block enclosed in single quotes and separated by a comma.

**clear to send (CTS):** An indication by the 5100 or I/O device that it is ready to receive data.

**command device address:** 1. The device address used for specifying the device characteristics. 2. Addresses an area of storage in the user storage where the user-specified device characteristics are stored.

**end-of-buffer code:** A code (hex FF) that indicates the end of the data in the 5100 input or output buffer.

**end of file:** A condition that exists when an input device does not have any more records to transmit to the 5100.

**end-of-record code:** A code (hex E3) that indicates the end of a record in the 5100 input or output buffer. When more than one record is in the buffer, this code is used to separate them.

**hexadecimal representation:** A number system with a base of 16. The hexadecimal digits are 0-9 and A-F, where A represents the decimal value 10, B represents 11, and so on.

**I/O device:** A device attached to the 5100 that transfers data to the 5100 or from the 5100.

**ignore mode:** A mode of operation where the 5100 does not check for a request to send before indicating whether or not it is ready to receive data.

**input device:** A device attached to the 5100 that transfers data to the 5100.

**input end-of-block character:** 1. A character transmitted from an input device that indicates a complete block of data was received by the 5100. 2. Indicates the end of transmission from the input device.

**input new-line character:** A character transmitted from an input device that indicates a complete record was transmitted to the 5100.

**input operation:** Transmitting data to the 5100 from an external I/O device.

**input operation:** Transmitting data to the 5100 from an external I/O device.

**input prompting character:** A special character automatically transmitted by the 5100 to indicate it is ready to receive data. This character tells the input device when to start sending data.

**long space:** See *break*.

**modem mode:** 1. A mode of operation where the 5100 simulates a modem and indicates it is ready to receive data after the I/O device makes a request to send data. The 5100 continues to indicate it is ready to receive data until the I/O device stops the request to send data.

**nonprompting mode:** A mode of operation where the 5100 does not automatically transmit an input prompting character when it is ready to receive data.

**numeric mode:** A mode of operation where the input data is not automatically blocked into 18 characters. The necessary delimiters must be provided as part of the input data.

**output device:** A device attached to the 5100 that transfers data from the 5100.

**output end-of-block character:** A character with a special meaning when transmitted to an output device; for example, to indicate a carriage return or line feed.

**output new-line character:** A character with a special meaning when transmitted to an output device; for example, to indicate a carriage return or line feed.

**output operation:** Transmitting data from the 5100 to an external I/O device.

**prompting mode:** A mode of operation where the 5100 automatically transmits an input prompting character when it is ready to receive data.

**request to send (RTS):** An indication by the 5100 or I/O device that it is ready to send data.

**set mode:** A mode of operation where the 5100 sets request to send or clear to send on and ignores any request to send or clear to send indications from the I/O device.

**terminal mode:** A mode of operation where the 5100 simulates a terminal and requests to send data and then begins sending data when the I/O device indicates it is ready to receive the data.

**translation tables:** Tables used to convert 5100 characters into 5-, 6-, or 7-bit codes, and vice versa.

**user storage:** The internal storage available for user-defined programs and data.

)CONTINUE command 35

)MODE COM 3

( $\Phi$ )character 8, 11

, function 48

Ⓒ character 50, 65

Ⓓ character 50, 65

adapt the 5100 3

APL shared variable 3, 11, 14, 22, 24, 42

APL translation table 36, 40

ASCII characters 1, 84

atomic vector 48

ATTN key 58, 91

    APL operations 42, 43

    BASIC operations 53, 54

available user storage 3

BASIC source program 18, 58

BASIC translation table 36, 40

Baudot character 1, 77

binary character format 1

break 69, 83

buffer 14, 42, 53

buffer size 15, 25, 27, 40

cables 72

catenate function 48

character mode 18, 27, 40, 54, 57

characteristics 69

clear to send (CTS) 12, 30, 69

clear to send signal 70

close the command device 55

closing the I/O device 42, 47, 57

CMD 0 43, 54, 58, 69

command device 3, 22, 24, 44, 55

communications feature 1

conventions 30, 69

conversion chart

    5-bit code 77

    6-bit code 80

    7-bit code 84

convert 6, 10

copy a stored workspace 42

CTS (see clear to send)

data communications equipment 72

data rate 26, 40, 69

data set 70

data set ready signal 70

data terminal equipment 72

data terminal ready 43, 54, 69

data terminal ready signal 70

data transferred immediately 22, 42, 49

default value 40, 44, 55

device address summary 88

device addresses 22, 44, 45, 55, 56

device characteristic summary 88

device characteristics 2, 15, 22, 24, 44, 55

disable actions taken by 5100 17, 21, 34

disable input end-of-block character 17, 34

disable input new-line character 17, 34

disable output end-of-block character 21, 35

disable output new-line character 21, 35

EBCD characters 1, 80

EBCD code 50, 65

EBCD control characters 83

EIA RS232C specification 30, 69

enable input end-of-block character 17, 34

enable input new-line character 17, 34

enable output end-of-block character 21, 35

enable output new-line character 21, 35

end of buffer code 16, 17, 20, 25

end of file 69

end of file conditions

    APL 43

    BASIC 54, 58

end of record code 16, 17, 19, 25

end of transmission 16

error messages 91

ERROR 002 ddd 92

ERROR 003 ddd 92

ERROR 004 ddd 94

ERROR 010 A02 54

ERROR 010 ddd 94

ERROR 013 ddd 94

ERROR 014 ddd 94

ERROR 152 56

establish a shared variable 44

even parity 39, 40

EXCEEDS MAXIMUM RECORD LENGTH 45

exchange data format 44, 45

external I/O device 1, 2

function key 54

general exchange data format 44, 45  
GET statement 4, 15, 23, 27, 57

half duplex operation 69  
hexadecimal character 9  
hexadecimal constants 9  
hexadecimal representation 9, 10

I/O device 1, 2, 22  
I/O device requirements 3  
IBM 2741 communications terminal 50, 65  
ignore mode 30, 40, 69  
IMFs 13  
immediate delay 36, 40, 51, 66  
input data flow  
    APL 48  
    BASIC 63  
input device 22, 23  
input end of block character 11, 16, 33, 40, 46, 63  
input end of buffer condition 17  
input end of data condition 17  
input end of record condition 17  
input new line character 11, 16, 33, 40, 46, 63  
input operation 3, 12, 14, 22, 23  
input operations  
    APL 42, 43, 46  
    BASIC 53, 56, 57  
input parity 39, 40  
input process arrow 12  
input prompting character 29, 40  
input translation table 38, 40  
internal machine fixes (IMFs) 13

keyboard 48, 62

list a BASIC program 58  
load a BASIC program 53, 58  
load a stored workspace 42  
LOAD command 15, 23, 58  
long-space 69, 83  
loss of data 48, 63  
lost data 48, 63  
lower shift 79, 83

modem 69, 70  
modem cable 30, 72  
modem mode 30, 40  
MSG=OFF 44, 45

no parity checking 39, 40  
nonprompting mode 28, 40  
numeric mode 27, 40

odd parity 39, 40  
OPEN statement 22, 55, 56  
opening the command device 3, 22, 44, 55  
opening the I/O device 3, 22  
    APL 45  
    BASIC 56  
option menu 3  
originate signal 70  
output data flow  
    APL 49  
    BASIC 64  
output device 22, 23  
output end of block characters 19, 21, 33, 40  
output end of buffer conditions 20  
output end of data condition 20  
output end of record condition 20  
output new-line characters 19, 21, 33, 40  
output operation 3, 12, 14, 22, 23  
    APL 42, 46  
    BASIC 53, 56, 57  
output parity 39, 40  
output process arrow 12, 52, 67  
output translation table 37, 40

parity 69  
parity error 8  
physical end of buffer 17, 19, 20, 21  
pin assignment 69, 70  
PRINT statement 3, 11, 14, 23, 53, 57  
prompting mode 28, 40  
punch a BASIC program 58  
PUT statement 3, 11, 14, 23, 53, 57

receive 69  
receive interrupt 69  
receive timeout 31  
received 6  
received data signal 70  
receiving data 1, 11, 12, 46  
records 14  
request to send (RTS) 12, 30, 69  
request to send signal 70  
retract 47  
retracting the shared variable 47  
return code 31, 44, 46  
RTS (see request to send)

sample APL program 50  
sample BASIC program 65  
SAVE command 14, 23, 53, 58  
serial I/O adapter program 3  
serial I/O adapter self test 3, 72, 73  
serial I/O modem attach 72  
serial I/O terminal attach 72  
set mode 30, 40, 69  
set up procedure 63  
space 69  
special characters 48, 59, 62  
specifications 69  
specify device characteristics 22, 24, 44, 55  
stop bit error 8, 11  
stop bits 39, 40  
stop processing 36  
summary 40

terminal 69, 70  
terminal cable 30, 69, 70, 72  
terminal mode 30, 40, 69  
test option 3, 13  
timing requirements 12  
transferring data 46  
translation tables 7, 37, 38, 40  
    5-bit code 8  
    6-bit code 8  
    7-bit code 8  
    8-bit code 10  
transmit 69  
transmitted 6  
transmitted data signal 70  
transmitting data 1, 11, 46  
TYPE= 44

untranslatable character 8  
upper shift 79, 83  
user specified device characteristics 24, 44, 55  
user storage 3, 14  
UTIL MODE COM 3

wait mode 30

5-bit code 1, 3, 6, 8  
6-bit code 1, 3, 6, 8  
7-bit code 1, 3, 6, 8  
8-bit code 1, 3, 9, 15, 58  
9 0 return code 31, 46  
5100 keyboard 48, 62



READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). No reply.

Page Number    Error

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number    Comment

IBM 5100  
Serial I/O Adapter Feature  
Users Manual

SA21-9239-1

☐ Check if reply is requested.

Note: All comments and suggestions become the property of IBM.

Name \_\_\_\_\_

Address \_\_\_\_\_

● No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold

Fold

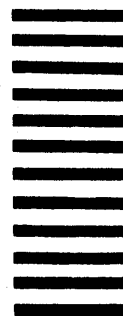
FIRST CLASS  
PERMIT NO. 387  
ROCHESTER, MINN.

**BUSINESS REPLY MAIL**

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation  
General Systems Division  
Development Laboratory  
Publications, Dept. 245  
Rochester, Minnesota 55901



Fold

Fold

IBM 5100 Serial I/O Adapter Feature Users Manual (Printed in U.S.A.)

SA21-9239-1



International Business Machines Corporation  
General Systems Division  
5775D Glenridge Drive N.E.  
Atlanta, Georgia 30301  
(USA Only)

IBM World Trade Corporation  
821 United Nations Plaza, New York, New York 10017  
(International)

C

O

C

C

O

O

O



International Business Machines Corporation  
General Systems Division  
5775D Glenridge Drive N.E.  
Atlanta, Georgia 30301  
(USA Only)

IBM World Trade Corporation  
821 United Nations Plaza, New York, New York 10017  
(International)