IBM

IBM 5100
Communications/Serial I/O
Maintenance Information Manual

5100

Preface

This manual is for personnel who service the IBM 5100
Portable Computer Communications Adapter feature,
Serial I/O Adapter feature, or both. To use this manual,
you should know how to operate the 5100 with the Com-
munications Adapter feature, Serial I/O Adapter feature, or
both. It would also be helpful if you understand basic tele-
processing concepts.

Related Publications

- *IBM 5100 Communications Adapter Feature User's
  Manual,* SA21-9215

- *IBM 5100 Serial I/O Adapter Feature User's Manual,*
  SA21-9239

- *IBM 5100 Maintenance Information Manual,*
  SY31-0405

Contents

Because the Communications Adapter feature and the Serial I/O Adapter feature are similar, both features are explained in this manual.

The main sections of the manual and what each explains is as follows:

- *Introduction* explains the similarities and the major differences between the two features, what each feature consists of, and the function of each feature.

- *Theory of Operation* explains the microprograms and data flow of each feature, the data format used with both features, the EIA (Electronics Industry Association) RS232C/CCITT V.24-V.28 standard interface lines that both features use, and how the expansion feature card functions in both transmit and receive operations.

- *Diagnostic Aids* tells what diagnostic aids are available for communications and serial I/O. This section also contains instructions on how to use the diagnostic buffer and network analysis routine for communications, and contains a troubleshooting guide for communications and serial I/O.

- *Communications Networks* explains some of the basics of data communications that apply to the IBM 5100 Portable Computer.

- *Appendix. Character Code Charts* shows the character codes and the character bit values that are used by both features.

The Serial I/O Adapter feature, the Communications Adapter feature, or both, can be installed on the 5100 Portable Computer.

The similarities of the communications and serial I/O Adapter features are:

• They both use the EIA (Electronics Industry Association) RS232C/CCITT V.24-V.28 standard interface to connect the 5100 to a data set or an I/O device.

• They both use the same logic card (B2).

• They both use microprograms in read/write storage.

The major differences between the two features are:

• The APL or BASIC language is active with the Serial I/O Adapter feature and a user written APL or BASIC program interacts with the serial I/O program. The APL or BASIC language is not active with the Communications Adapter feature.

• The features use a separate connector on the rear panel of the 5100.

EXPANSION FEATURE

The expansion feature is a prerequisite for the Communications Adapter feature, the Serial I/O Adapter feature, or both. The expansion feature is the expansion feature card (B2).

COMMUNICATIONS ADAPTER FEATURE

The Communications Adapter feature enables the IBM 5100 Portable Computer to communicate with a remote system. The 5100 appears the same as an IBM 2741 Communication Terminal to the remote system. The 5100 can transmit data from the keyboard or tape unit, write transmitted or received data on tape, or print transmitted or received data.

The Communications Adapter feature consists of internal and external cables, and the communications program (the expansion feature is also required). The communications program, which consists of the communications microprogram, the communications I/O microprogram, and the printer microprogram, is stored on tape and must be loaded into read/write storage before using the feature. For detailed information on loading the communications program and using the feature, refer to the *IBM 5100 Communications Adapter Feature User's Manual.*

SERIAL I/O ADAPTER FEATURE

The Serial I/O Adapter feature allows a user to attach I/O devices other than the 5100 I/O devices to the 5100 Portable Computer.

The Serial I/O Adapter feature consists of internal and external cables, and the serial I/O program (the expansion feature is also required). The serial I/O program, which consists of the serial I/O microprogram and the serial I/O input/output microprogram, is stored on tape and must be loaded into read/write storage before using the feature. For an explanation of how to load and use the serial I/O program, refer to the *IBM 5100 Serial I/O Adapter Feature User's Manual.*

COMMUNICATIONS MICROPROGRAMS AND
DATA FLOW

The following diagram shows the location of the micro-
programs and the data flow for a communications transmit
and receive operation. Refer to this diagram while reading
the description of the microprograms and data flow.

Read/Write Storage

Extended Display	Display Screen

Executable ROS

Printer	Printer Microprogram (option 2 or 4)	Communications Microprogram
	Communications Diagnostic Buffer	Communications I/O Microprogram

I/O Supervisor Microprogram	Printer Microprogram (option 1 or 3)	Printer
Microprogram Subroutines	Tape Microprogram (option 1 or 3)	Tape
	Keyboard Microprogram	Keyboard

Expansion Feature Card	Data Set

Receive - - - - -

Transmit ————
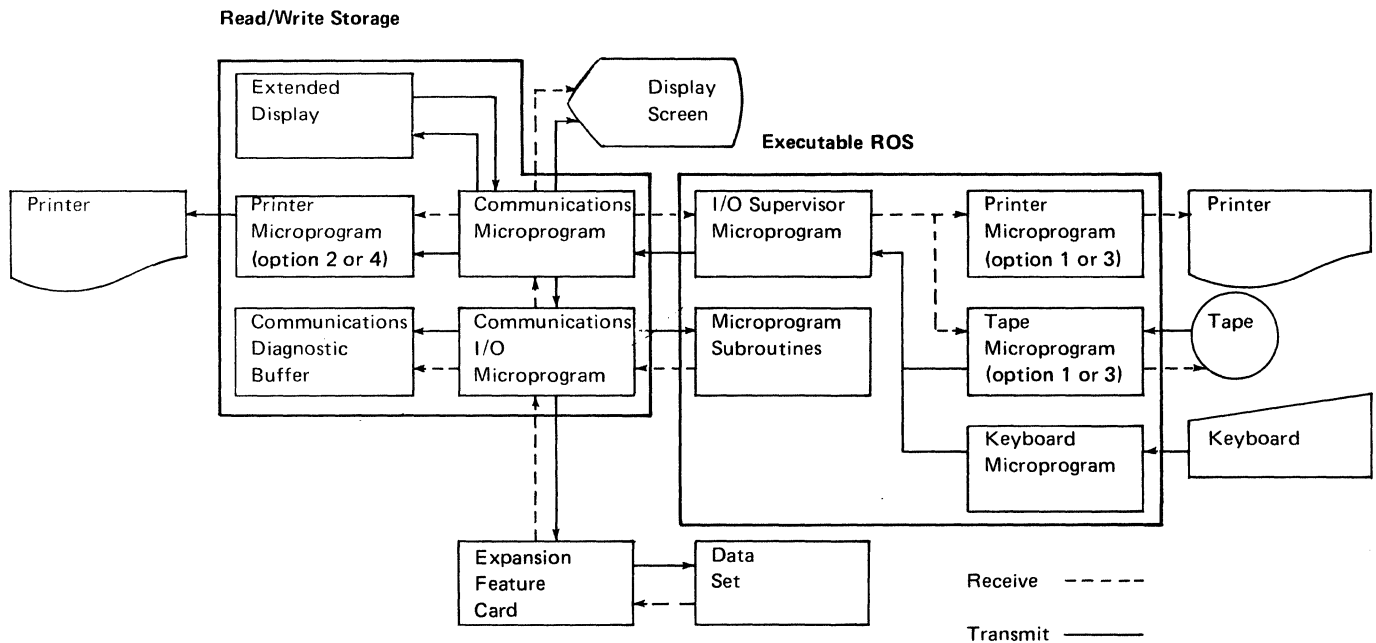
During communications, the IBM 5100 Portable Computer
functions as a communications terminal, and normal APL
and BASIC language operations cannot be performed.

During communications, the 5100 uses microprograms in
read/write storage and microprograms and subroutines in
executable ROS.

The microprograms in read/write storage that are used
during communications are the communications micro-
program, the communications I/O microprogram, and the
printer microprogram (used only when option 2 or 4 is
selected).

The microprograms in executable ROS that are used during communications are:

● The printer, tape, keyboard, and I/O supervisor (these are also used in normal APL and BASIC operations).

● The microprogram subroutines (these are used only by the communications I/O microprogram).

The microprograms in read/write storage are loaded from the communications tape; however, the printer microprogram is loaded only when option 2 or 4 is selected. During APL or BASIC operations, or if option 1 or 3 is selected, the printer microprogram in executable ROS is used for printer operations. Refer to the *IBM 5100 Communications Adapter Feature User's Manual* for a description of the options.

The communications microprogram controls the format of the data printed and written on tape, stored in the extended display, and displayed on the screen. It also adds the line control and function characters, and transfers data to or from the communications I/O microprogram.

Transmit

During transmit operations, data can be supplied from the keyboard, the extended display, or a tape. The communications I/O microprogram converts the data to EBCD (extended binary coded decimal) or correspondence code characters, places the character in the communications diagnostic buffer, serializes each character, adds start and stop bits, and sends the character to the expansion feature card one bit at a time.

The operating sequence of the microprograms and the data flow is as follows:

1. A key is pressed on the keyboard or a character is read from tape.

2. The keyboard or tape microprogram in executable ROS passes the data from the keyboard or tape to the I/O supervisor.

3. The I/O supervisor passes the data to the communications microprogram.

4. The communications microprogram places the character (if it is a data character) in the extended display and on the display screen. If the 5100 is transmitting and the character was entered from the keyboard, it is passed to the communications I/O microprogram. If the character was read from tape or the 5100 is in the edit state, the character is not passed to the communications I/O microprogram until a complete line is read from tape or until the EXE-

CUTE or ATTN key is pressed. The data is then read from the extended display; control and function characters are added; and the data is passed to the communications I/O microprogram one character at a time.

5. The communications I/O microprogram, with the use of the microprogram subroutines in executable ROS, converts the character to EBCD or correspondence code; places the character in the diagnostic buffer; adds the start and stop bits; and sends the character, bit by bit, to the expansion feature card. The bits sent to the card are then sent to the data set and transmitted to the remote system.

Receive

All data received by the 5100 is stored in the extended display, displayed on the screen, and printed and/or written on tape. The communications I/O microprogram removes the start and stop bits from each character, deserializes the character, places the character in the communications diagnostic buffer, converts the character to EBCD or correspondence code and sends it to the communications microprogram.

The operating sequence of the microprograms and the data flow when the 5100 is receiving is:

1. Bits from the data set are sent to the expansion feature card.

2. The expansion feature card sends the bits on bus in to the communications I/O microprogram.

3. The communications I/O microprogram assembles the bits into EBCD or correspondence code characters, updates the communications diagnostic buffer with the characters, and passes control to the communications microprogram.

4. The communications microprogram updates the display screen and extended display, then waits for the next character.

5. If option 2 or 4 is selected, the data is printed when a line feed or new-line function character is received. If option 1 or 3 is selected, the data is stored in the extended display until a new-line character and the end of transmission character (Ⓒ) are received, then printed or written on tape.

6. The data flow from the extended display to the printer or tape is controlled by the communications microprogram, the I/O supervisor, and the tape and printer microprograms.

SERIAL I/O MICROPROGRAMS AND DATA FLOW

The following diagram shows the location of the serial I/O
microprograms and the data flow for a serial I/O transmit
and receive operation. Refer to this diagram while reading
the description of the microprograms and data flow.
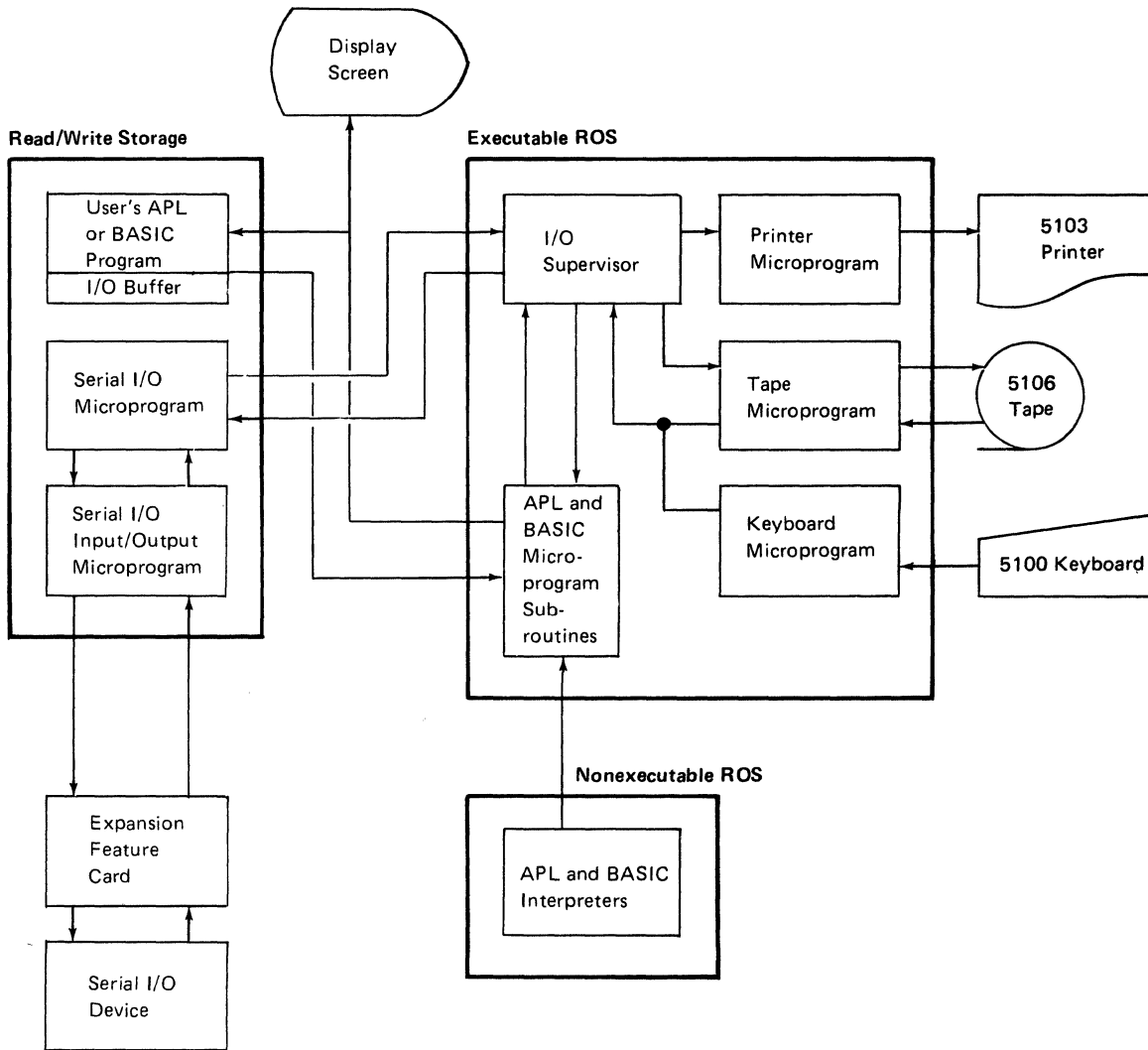
During serial I/O operations, the 5100 uses programs in
read/write storage, executable ROS, and nonexecutable
ROS.

The programs in read/write storage are the user's APL or
BASIC program, the serial I/O microprogram, and the
serial I/O input/output microprogram.

The programs in executable ROS are the printer, tape,
keyboard, and I/O supervisor microprograms. The APL
and BASIC microprogram subroutines in executable ROS
are also used during serial I/O operations.

The programs in nonexecutable ROS are the APL and BASIC interpreters.

The microprograms in executable and nonexecutable ROS function the same in serial I/O operations as they do in normal APL or BASIC language operations.

The serial I/O microprograms in read/write storage are loaded from the serial I/O adapter tape cartridge.

The serial I/O microprogram sets the device characteristics and the request to send and clear to send controls as specified by the user. (See the *IBM 5100 Serial I/O Adapter Feature User's Manual* for an explanation of how these device characteristics are set.) This microprogram also controls the data flow into and out of the I/O buffer in read/write storage. The size of the I/O buffer is set by the serial I/O microprogram as specified by the user. (See the *IBM 5100 Serial I/O Adapter Feature User's Manual* for more information on the I/O buffer.)

During a transmit operation, the serial I/O microprogram takes the data from the I/O buffer when the device (output file) is closed or when the buffer becomes full. This microprogram also adds control and function characters if necessary, such as upper shift function characters, and end-of-block and end-of-buffer control characters; and sends the data to the serial I/O input/output microprogram one character at a time.

During a receive operation, the serial I/O microprogram puts the data in the I/O buffer as it is received from the I/O device.

Transmit Operation

During a transmit operation, data is supplied by the user's APL or BASIC program. (During BASIC language operations, the user can transmit a BASIC program to the I/O device with the SAVE command.) The serial I/O input/output microprogram converts the data to the specified code (5-, 6-, 7-, or 8-bit), adds the start and stop bits, serializes each character, and sends the character to the expansion feature card one bit at a time.

The operating sequence of the microprograms and the data flow during a transmit operation is:

1. A data character is entered from the keyboard or read from tape.

2. The keyboard or tape microprogram passes the data character to the I/O supervisor.

3. The I/O supervisor passes the data character to the I/O buffer in read/write storage via the APL or BASIC microprogram subroutines, which use the programs in the APL or BASIC interpreters.

4. When the user's program specifies output, the serial I/O microprogram removes the data from the I/O buffer via the I/O supervisor and APL or BASIC microprogram subroutines; adds any necessary function and control characters; and passes the data to the serial I/O input/output microprogram.

5. The serial I/O input/output microprogram converts the character to the specified code, serializes the character, adds start and stop bits, and sends the character, bit by bit, to the expansion feature card. The bits sent to the card are transmitted to the serial I/O device on the transmitted data line.

Note: Steps 1, 2, and 3 are the same as a normal APL or BASIC language operation. The data could also be supplied by the user's APL or BASIC program, such as the result of calculations.

Receive Operation

During a receive operation, data received from an I/O device is sent to the user's APL or BASIC program by the serial I/O microprograms; the user's program may use the data or send it to the 5100 output devices. (In BASIC language operations, the user can receive a BASIC program from the I/O device with the LOAD command.)

The operating sequence of the microprograms and the data flow during a receive operation is:

1. The 5100 receives the data from the I/O device on the received data line via the expansion feature card.

2. The card sends the data, bit by bit, to the serial I/O input/output microprogram on bus in.

3. The serial I/O input/output microprogram deserializes each character, removes the start and stop bits, converts the character to 5100 code, and sends the character to the serial I/O microprogram.

4. The serial I/O microprogram places the character in the I/O buffer via the APL or BASIC microprogram subroutines and the I/O supervisor.

5. The user's program uses the data or sends it to the 5100 I/O devices via the APL or BASIC microprogram subroutines and the I/O supervisor as in a normal APL or BASIC operation.

DATA FORMAT

The 5100 uses the start-stop format for both the communications and serial I/O features. In the start-stop format, each character is synchronized separately on the data line. To synchronize each character, a ST (start) bit is added to the beginning of the character and a SP (stop) bit(s) is added to the end of the character. The data bits transmitted between the ST and SP bits can be any one of the codes used with the 5100 communications or serial I/O feature, because the ST and SP bits, respectively, indicate the beginning and end of a character.

The code used with the communications feature is EBCD (extended binary coded decimal) or correspondence code. The code used with the serial I/O feature is 5-, 6-, 7-, or 8-bit code.

5-Bit Code

Each 5-bit character contains 5 data bits labeled 1, 2, 3, 4, and 5. There are no parity bits with the 5-bit code. The 5-bit code is compatible with I/O devices that use the Baudot code.

Additional characters are obtained by preceding the code with a letters (lower) or figures (upper) shift character. If the characters are preceded by a letters character, the characters are letter (lower) shift characters. If the characters are preceded by a figures character, the characters are figure (upper) shift characters. For more information on letter and figure characters, see the *IBM 5100 Serial I/O Feature User's Manual.*

6-Bit Code

Each EBCD, correspondence code, or 6-bit character contains 7 bits, which are labeled B, A, 8, 4, 2, 1, and C. Bits B, A, 8, 4, 2, and 1 are data bits and indicate different characters by using a different combination of bits for each character. Bit C is a check bit (parity bit) and is used to maintain odd parity of each character for error checking purposes. The 6-bit code is compatible with I/O devices that use the EBCD or correspondence code.

Additional character combinations are obtained by preceding a character or characters with a shift function character. For example, if the characters ABCD are preceded by an upper shift function character, the characters ABCD are upper shift characters. For more information on shift function characters during communications operations, see *Shift Function Characters* and *Communications Diagnostic Buffer.* For more information on shift function characters during serial I/O operations, see the *IBM 5100 Serial I/O Feature User's Manual.*

7-Bit Code

Each 7-bit character contains 8 bits, which are labeled P, 1, 2, 3, 4, 5, 6, and 7. Bits 1, 2, 3, 4, 5, 6, and 7 are data bits and bit P is a parity bit. The parity of a 7-bit character can be odd, even, or ignored as specified by the user. When the ignore option is specified, all 7-bit characters that the 5100 transmits to the I/O device have the parity bit turned off (space level); and on all 7-bit characters that the 5100 receives from the I/O device, the parity bit is ignored. The 7-bit code is compatible with I/O devices that use ASCII (American Standard Code for Information Interchange).

8-Bit Code

Each 8-bit character contains 8 data bits (no parity bits), which are labeled 0, 1, 2, 3, 4, 5, 6, and 7. The 8-bit code uses the hexadecimal format 00 through FF.

See the Appendix, *Character Code Charts* for the bit formats and character graphics for the EBCD, correspondence, 5-, 6-, and 7-bit codes.

The presence of a data bit on the received data or transmitted data line is indicated by a mark; a mark is a minus (−) voltage level (−3.0 to −25.0V). The absence of a data bit is indicated by a space; a space is a plus (+) voltage level (+3.0 to +25.0V). The following graph shows how the character A would look on the data line in each code with the ST and SP bits added:

First Bit Transmitted — Increasing Time — Last Bit Transmitted

Space + ST 1 2 3 4 5 SP
Mark −
1-1/2 Bits

5-Bit Character A (bits 1 and 2) No Parity

Space + ST B A 8 4 2 1 C SP
Mark −

EBCD or 6-Bit Character A (bits B, A, and 1) Odd Parity[1]

Space + ST 7 6 5 4 3 2 1 P SP[2]
Mark −

7-Bit Character A (bits 7 and 1) Even Parity

Space + ST 7 6 5 4 3 2 1 0 SP[2]
Mark −

8-Bit Character A (bits 7, 6 and 0) No Parity

---
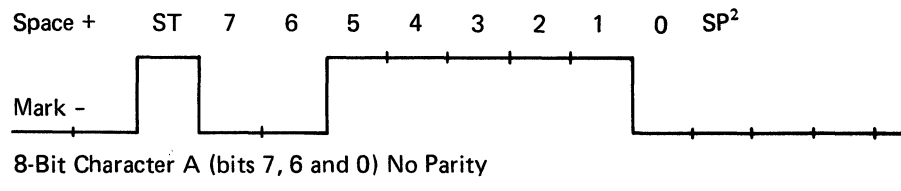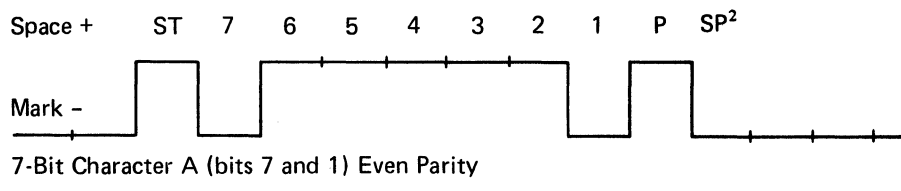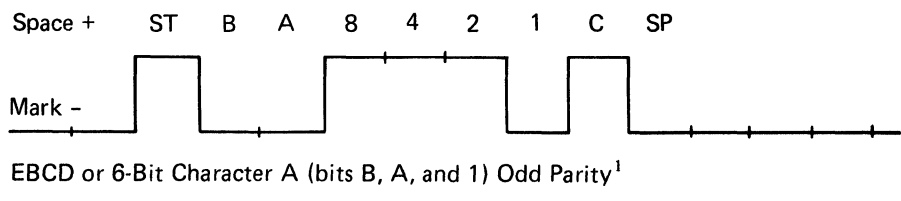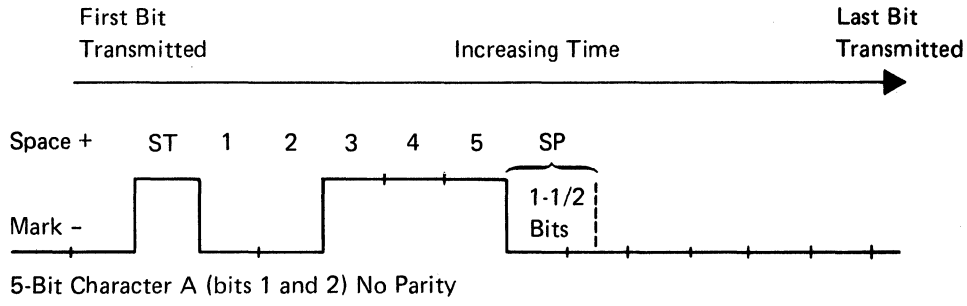
[1] For Correspondence code, the bit value is B, 4, 2, 1, and C, (see *Code Chart for Correspondence Code Data Characters*).

[2] Can have two SP bits if specified by the user.

COMMUNICATIONS CONTROL CHARACTERS

Messages are indicated by two line control characters: Ⓓ (circle D) and Ⓒ (circle C). A Ⓓ indicates the start of a message, or start of transmission; a Ⓒ indicates the end of a message, or end of transmission. These characters are transmitted by the 5100 and by the remote system as shown in the following diagram (the arrow indicates the direction of data flow):

Action	IBM 5100	Remote System

The lines 'clear to send' and 'data set rdy' are activated by the data set, which is indicated by the arrow on the status line of the display screen pointing up ↑.

The 5100 waits approximately 1 second after 'clear to send' and 'data set rdy' become active, then sends the first message to the remote system.

Ⓒ . . . data Ⓓ [1] →
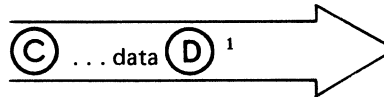
The 5100 waits for the remote system to send a message.

← Ⓓ data . . . Ⓒ
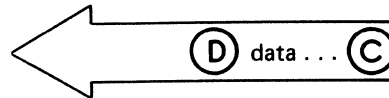
The remote system responds.

After a 66-millisecond delay, the 5100 changes to the transmit state, and transmits the next message.

Ⓒ . . . data Ⓓ →

---
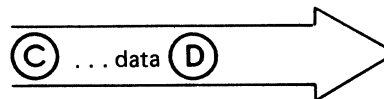
[1] The Ⓓ is transmitted by the communications program immediately after the approximate 1-second delay, which is indicated by the arrow on the status line changing from up (↑) to right (→).

If the remote system does not respond to a message from the 5100 or if one of the control characters is lost (because of line noise or some other reason) the 5100:

- Remains in the receive state until a Ⓒ or Ⓓ is received.

- Ignores all data characters until a Ⓒ or a Ⓓ is received.

- Goes to the transmit state when a Ⓒ is received before a Ⓓ. Any data characters received before the Ⓒ is received are lost.

- Remains in the receive state if a Ⓓ is received but no Ⓒ is received. (Any data received is stored in the extended display but is not printed or written on tape.) The 5100 leaves the receive state when either the ATTN key is pressed (which transmits a receive interrupt to the remote system) or the CMD and ATTN keys are pressed (which causes the 5100 to go to the transmit state and send a Ⓓ).

COMMUNICATIONS FUNCTION CHARACTERS

Function characters are special characters that the 5100 and the remote system use during communications to control specific functions. These characters, their function, and their bit values, are shown under *Code Chart of Function and Control Characters* in the Appendix, *Character Code Charts.*

Following is a more detailed explanation of the function characters that apply to the 5100:

Shift Function Characters

The shift function characters, upper shift (indicated by a < in the diagnostic buffer) and lower shift (indicated by a ≤ in the diagnostic buffer) indicate to the remote system and the 5100 that the data following is upper or lower shift (see *Communications Diagnostic Buffer*). For example, if the 5100 received:

   < ABCDEF ≤GHIJK

the characters ABCDEF would be upper shift characters and the characters GHIJK would be lower shift characters as indicated by the preceding shift function character.

Backspace Function Character

A backspace function character (indicated by a $\bar{n}$ in the diagnostic buffer) indicates to the 5100 and the remote system to backspace one character. For example, if the 5100 received:

   A$\bar{n}$ < F

the 5100 would display the character A, backspace the cursor, and display the character _ (an upper shift F) to form the character A̲.

Tab Function Character

A tab function character (indicated by a ⊂ in the diagnostic buffer) indicates to the 5100 and remote system to tab to the next tab setting. For example, if the 5100 received:

   ABCD ⊂EFGH

the display would show:

   ABCD   EFGH

The number of spaces would depend on where the tab was set.

Bypass and Restore Function Characters

The bypass and restore function characters (indicated by a ω and a ⊃ respectively in the diagnostic buffer) set and restore the bypass function on the 5100 when the characters are received. When the bypass function is set, (bypass function character received) the 5100 does not display or print data. If a tape file is open (&OPEN command is entered), the data is written on tape.

The bypass function is used for data security. Characters entered from the keyboard are sent to the remote system and the cursor advances with each key pressed.

The 5100 does not transmit the bypass and restore function characters.

New-Line and Line Feed Function Characters

A new-line function character (indicated by a ⊙ in the diagnostic buffer) indicates the end of a line of data and starts a new line of data. For example, if the 5100 received:

ABCDE⊙FGHI⊙

the display would show:

ABCDE

FGHI

—   (cursor)

A line feed function character (indicated by a ≠ in the diagnostic buffer) is only received by the 5100; the 5100 does not transmit this character.

A line feed function character indicates the end of a line of data and causes the display to move up one line (the cursor does not move to the left). For example, if the 5100 received:

ABCD≠EFGH≠

the display would show:

ABCD

    EFGH

       —   (cursor)

New-line and line feed function characters also indicate to the 5100 to start printer or tape output operations. However, only complete lines of data are printed and/or written on tape. A complete line is data ending with a new-line or line feed function character. For example, if the following message is received:

(D) ABC⊙DEF⊙GHI (C)

New-Line Characters
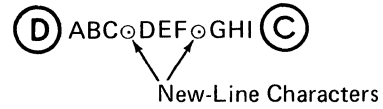
the last line of data (GHI) is not printed or written on tape because it does not end with a new-line character; however, it is stored in the extended display.

If the user enters the command & OUTSEL SYS, only
lines received from the remote system that end with a new-
line or line feed function character (complete lines) are
printed and/or written. A line of data (transmitted or re-
ceived data) without a new-line character is joined with
the following line. The following example shows when
data would be printed and/or written if the command &
OUTSEL SYS is entered and the messages are transmitted
as shown:

Note: The arrow indicates the direction of data flow.

IBM 5100	Remote System	5103 Printer and/or Tape Output	
(C) ⊙ ... TAP (D) ⟶		No output	(transmitted message)
⟵ (D) STEVE ... ⊙ (C)		STEVE	(received message)
(C) ... LEM (D) ⟶		No output	(received message without a new-line character)
⟵ (D) PAT ... ⊙ (C)		MELPAT	(received message with a new-line character)
(C) ⊙ ... NOD (D) ⟶		No output	(transmitted message)
⟵ (D) DAN ... (C)		No output	(received message without a new-line character)
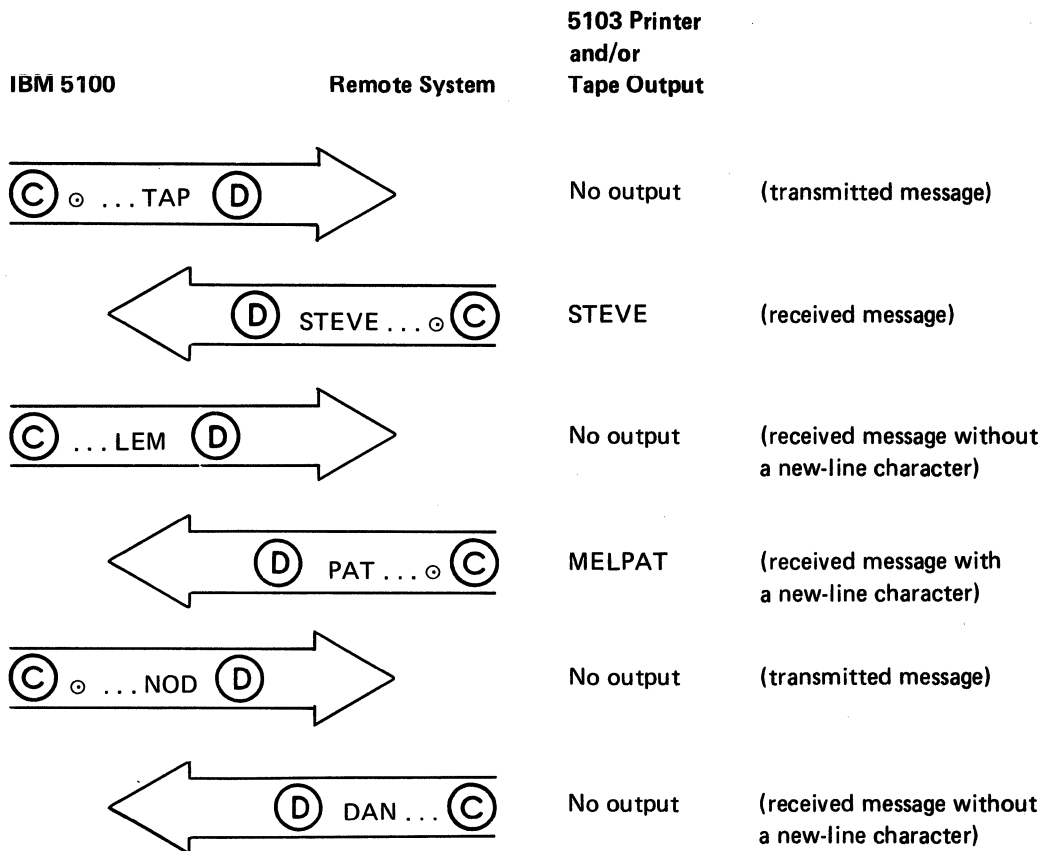
The message DAN is stored in the extended display, but is
not printed or written because a new-line or line feed func-
tion character was not received.

If communications continue and the extended display be-
comes full and overflows before a complete line of data
is received, the message DAN will be lost.

COMMUNICATIONS DATA SET EIA INTERFACE

The data set interface between the expansion feature card and the data set is the EIA RS232C/CCITT V.24-V.28 standard interface and is connected as shown in the following diagram:

Base I/O Card (F2)		Expansion Feature Card (B2)	Line Name	Pin No.	Data Set
	Bus Out →		Data Terminal Ready	20	
	Bus In ←		Request to Send	4	
	Device Adr Bus →		Transmitted Data	2	
	Control →		Received Data	3	
	'+ext horiz dr' (from display card) →		Clear to Send	5	
			Data Set Ready	6	
			Signal Ground	7	
			Protective Ground	1	
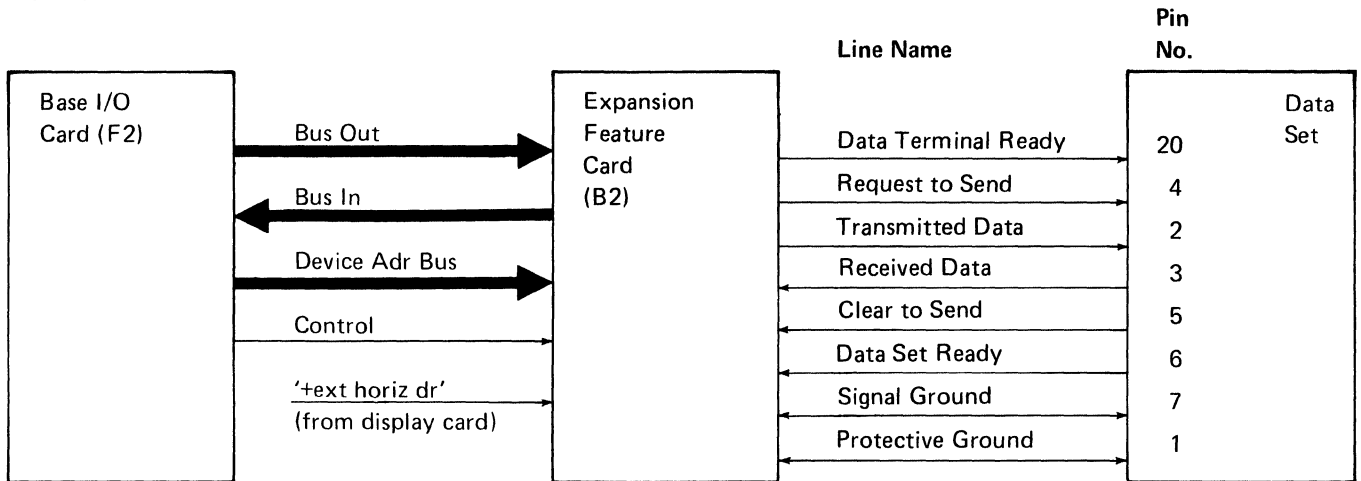
The output EIA lines are turned on, and the input EIA lines are sensed, by the communications I/O microprogram. An on signal on an EIA control line is indicated by a plus voltage level of +3.0 to +25.0V. An off signal is indicated by a minus voltage level of −3.0 to −25.0V.

The function of each EIA line and the sequence in which they are turned on or sensed during communications is:

If the 5100 is going to transmit data:

1. The 5100 turns on data terminal ready to indicate that it is ready to communicate.

2. The data set turns on data set ready to indicate that it is ready to communicate.

3. The 5100 turns on request to send to indicate that it is ready to send data to the data set.

4. The data set turns on clear to send to indicate that it is ready to send data to the remote system.

5. The 5100 sends the data to the data set on the transmitted data line. (See *Data Format* in *Theory of Operation*.)

If the 5100 is going to receive data:

1. Data terminal ready and data set ready are turned on as in a transmit operation.

2. Clear to send and request to send are not used.

3. The 5100 receives the data from the data set on the received data line. (See *Data Format*.)

Signal ground provides a common ground for all circuits. Protective ground is connected to the frame of the 5100. Refer to the *IBM 5100 Maintenance Information Manual*, for the communications connector and expansion feature card pin numbers.

SERIAL I/O DEVICE EIA INTERFACE

The 5100 uses the EIA RS232C/CCITT V.24-V.28 standard interface to connect the I/O devices used with the Serial I/O Adapter feature.

The EIA lines on the expansion feature card are turned on (output lines) and sensed (input lines) by the serial I/O input/output microprogram. The control lines are at a plus voltage level when active (on) and a minus voltage level when inactive (off). The acceptable voltage levels are +3.0 to +25.0V for an on level and −3.0 to −25.0V for an off level. For the voltage levels and the format of the data on the data lines (transmitted data and received data), see *Data Format*.

When operating the serial I/O feature, the user can select
one of four modes (modem, ignore, terminal, or set). In
modem or ignore mode, the 5100 functions like a modem
(data set). In terminal mode, the 5100 functions like a
terminal and in set mode, the 5100 can function as either
a terminal or a modem (either the modem cable or the
terminal cable can be used to attach the I/O device).

Modem Mode

Because the 5100 operates like a modem (data set) in
modem mode and the expansion feature card is designed for
communications terminal operations, the line names are
interchanged at the serial I/O connector as follows:

Data terminal ready with data set ready
Request to send with clear to send
And transmitted data with received data

The following diagram shows the EIA standard interface
lines that are used during serial I/O operations and the way
they are connected to the 5100 in modem mode. Refer to
the *IBM 5100 Maintenance Information Manual* for the
serial I/O connector and Z1 cable socket pin numbers.
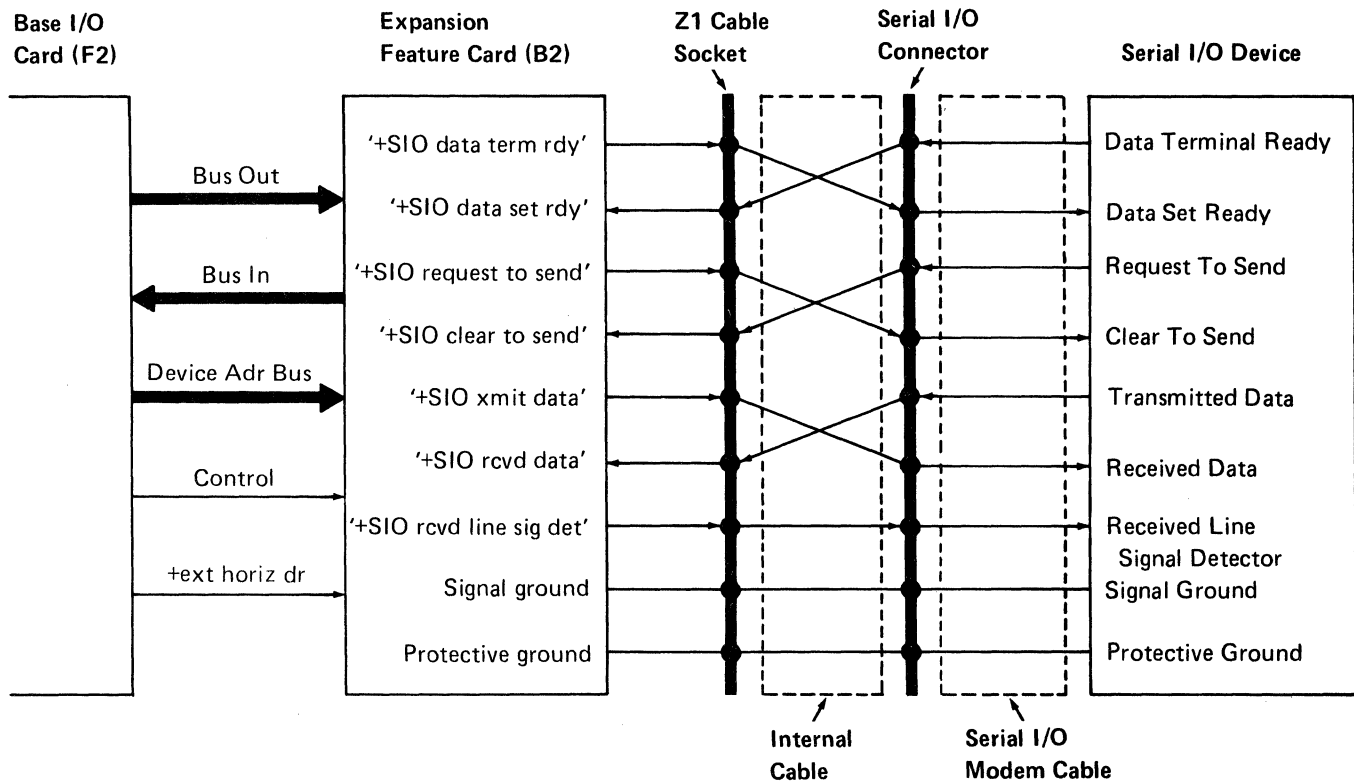
The function of each EIA line and the sequence in which
they are turned or sensed during serial I/O operations in
modem mode is:

- After the serial I/O program is loaded and enabled, the
  5100 turns on '+SIO data term rdy', which is sensed on
  the data set ready line at the I/O device, to indicate
  that the 5100 is ready for serial I/O operations.

- The serial I/O device turns on data terminal ready,
  which is sensed on the '+SIO data set rdy' line on the
  expansion feature card, to indicate that the serial I/O
  device is ready for serial I/O operations.

Note: The sequence in which data terminal ready and data
set ready are activated is not significant.

When the serial I/O device is going to transmit data to the
5100:

1. The serial I/O device turns on the request to send
    line, which is sensed at the '+SIO clear to send'
    line on the expansion feature card, to indicate that it
    is ready to send data.

2. The 5100 then turns on the '+SIO request to send'
    line, which is sensed on the clear to send line at the
    I/O device, to indicate that it is ready to receive the
    data from the I/O device.

3. The serial I/O device sends the data on the transmitted
    data line, which is received on the '+SIO rcvd data'
    line at the expansion feature card.

When the serial I/O device is going to receive data from the
5100:

1. The lines data terminal ready and data set ready are
    activated the same as when the I/O device is trans-
    mitting data to the 5100.

2. The line request to send from the I/O device ('+SIO
    clear to send' at the expansion feature card) must
    be off. Therefore, the line '+SIO request to send'
    at the card (clear to send at the I/O device) will also
    be off.

3. The 5100 sends the data on the '+SIO xmit data'
    line and the data is received on the received data line
    at the I/O device.

The received line signal detector line is active when the
'+SIO data term rdy' line is active and is provided for
I/O devices that require this signal.

Ignore Mode

In ignore mode, the sequence and function of the lines
are the same as they are in modem mode. (See *Modem
Mode*.) However, when the I/O device is going to send data
to the 5100, the 5100 turns on the '+SIO request to send'
(clear to send line at the I/O device) without checking the
request to send line from the I/O device ('+SIO clear to
send' at the expansion feature card).

When the serial I/O device is going to receive data from the
5100, the line '+SIO request to send' (clear to send at the
I/O device) remains off.

Terminal Mode

A flip cable (terminal cable) is used in terminal mode to
connect the serial I/O device to the serial I/O connector.
In this cable the following lines are interchanged:

   Data terminal ready and data set ready
   Request to send and clear to send
   Transmitted data and received data

Therefore, the line names at the serial I/O device conform
to the line names at the expansion feature card.

The following diagram shows the lines that are used during
serial I/O operations and the way they are connected to the
5100 in terminal mode. Refer to the *IBM 5100 Mainten-
ance Information Manual* for the serial I/O connector and
Z1 cable socket pin numbers.
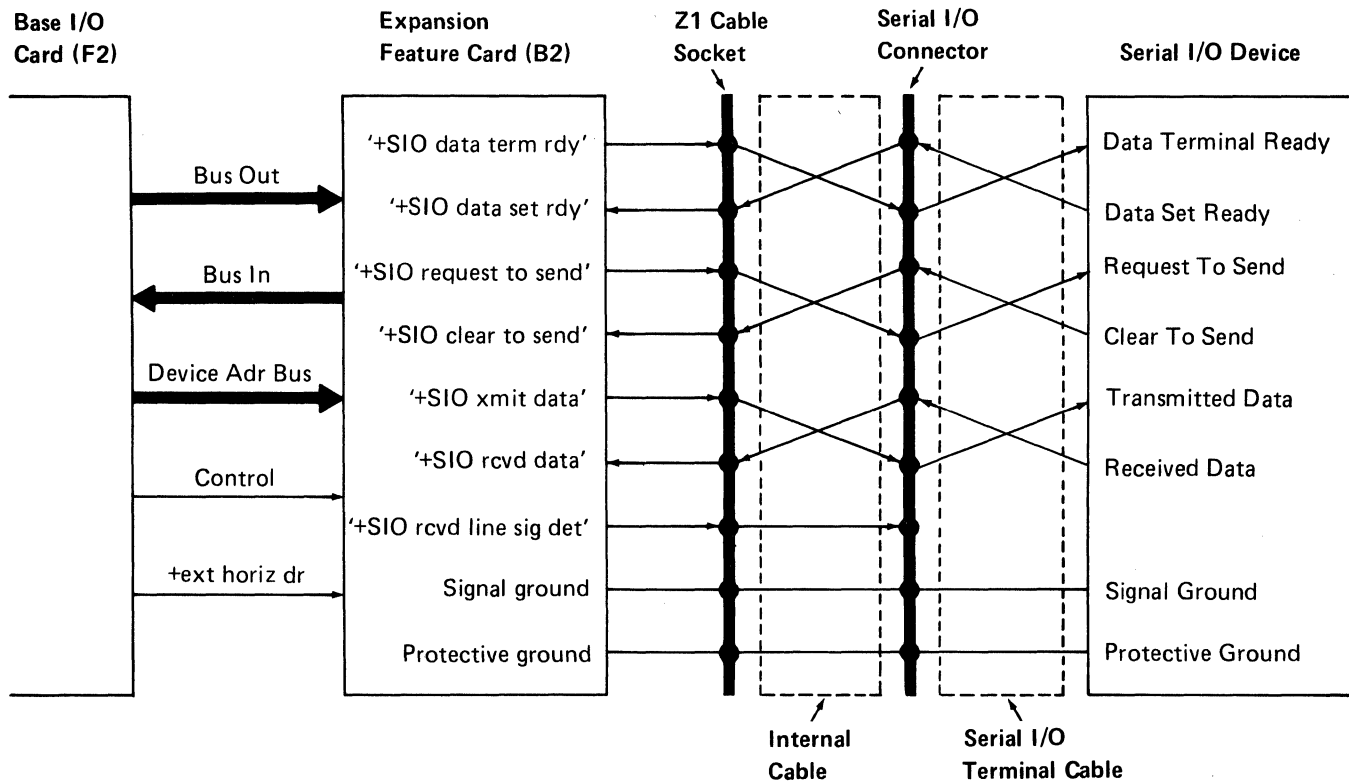
The function of each line and the sequence in which they
are turned on and sensed in terminal mode is as follows:

● After the serial I/O program is loaded and enabled, the
  5100 turns on '+SIO data term rdy' to indicate that it
  is ready for serial I/O operations.

● The serial I/O device turns on data set ready, which is
  sensed at '+SIO data set ready' at the expansion feature
  card.

Note: The sequence in which data terminal ready and data
set ready are activated is not significant.

When the 5100 is going to transmit data to the serial I/O
device:

1. The 5100 turns on '+SIO request to send' to indicate
    that it is ready to send data.

2. The serial I/O device turns on clear to send ('+SIO
    clear to send' at the expansion feature card) to
    indicate that the device is ready to receive data from
    the 5100.

3. The 5100 sends the data on the '+SIO xmit data'
    line to the serial I/O device.

When the 5100 is going to receive data from the serial I/O
device:

1. The lines data terminal ready and data set ready are
    activated the same as when the 5100 is transmitting
    data to the I/O device.

2. The line request to send ('+SIO request to send' at
    the card) remains off. Thus, the line '+SIO clear to
    send' will also be off.

3. The data is received from the serial I/O device on the
    '+SIO rcvd data' line.

2-14

Set Mode

In set mode, either the modem cable or the terminal cable can be used to connect the I/O device to the 5100.

When the modem cable is used, the sequence and function of the lines are the same as they are in modem mode. (See *Modem Mode*.) However, the 5100 turns on '+SIO request to send' (clear to send at the I/O device) without checking the request to send line ('+SIO clear to send' at the expansion feature card) from the I/O device. The '+SIO request to send' line remains on during transmit and receive operations.

When the terminal cable is used, the sequence and function of the lines are the same as they are in terminal mode. (See *Terminal Mode*.) However, the 5100 turns on '+SIO request to send' (request to send at the I/O device) and does not check the clear to send line ('+SIO clear to send' line at the expansion feature card) before sending data to the I/O device. The '+SIO request to send' line remains on during transmit and receive operations.

Note: The signal ground line provides a common ground for all circuits. The protective ground line is connected to the frame of the 5100.

EXPANSION FEATURE CARD

The expansion feature card is the interface between the base I/O card and the data set or serial I/O device. The card:

- Stores control information and transmit data that is sent by the communications or serial I/O microprogram

- Stores status information and received data that is read by the communications or serial I/O microprogram

- Provides interrupts to the communications or serial I/O microprogram at a selectable rate for receive data sampling and transmit timing

- Contains drivers and receivers to convert the data set and serial I/O device EIA voltage levels to 5100 voltage levels

- Checks the device address and bus out for correct parity when the expansion feature card is addressed

- Monitors the 'rcvd data' line during communications receive operations for a long space condition, and signals the processor via an interrupt when a long space occurs

- Switches between the communications and serial I/O connector

Refer to the expansion feature card logic diagram in the *IBM 5100 Maintenance Information Manual* while reading the following description of the expansion feature card operation.

Addressing

The expansion feature card is addressed by the '+device adr X3' and '+device adr Y3' (DAF) or '+device adr Y0' and '+device adr X2' (DA8) lines from the base I/O card on the device address bus.

DAF (device address F), '-control strobe', and '-bus out 0' reset the card logic.

DA8 (device address 8) addresses the adapter during functions other than reset.

Control Strobe

The '-control strobe' line is made active by the communications I/O microprogram or the serial I/O input/output microprogram to gate control information to the card. This control information is sent to the card on bus out and controls the card when the lines are active as follows:

'-bus out 0'

 The timer interrupt latch is allowed to be set so the card can generate a program level 1 interrupt.

'-bus out 2'

 The data rate clock starts when a start bit is received. If '-bus out 2' is inactive, the clock is allowed to run continuously.

'-bus out 3'

 The timer interrupt latch is reset.

'-bus out 4'

Selects communications circuitry. If '-bus out 4' is inactive, serial I/O circuitry is selected.

'-bus out 5'

The long space interrupt latch is reset.

'-bus out 6'

The '+com data term rdy' line to the data set, or the '+SIO data term rdy' line to the serial I/O device, and other controls within the card are set. When serial I/O, modem mode, is selected, '-bus out 6' sets the '+SIO rcvd line signal detector' to the serial I/O device.

'-bus out 7'

The control logic, which prevents noise from being accepted as a start bit, is turned on to detect the start bit of a character.

Put Strobe

The '-put strobe' line is activated by the communications I/O microprogram or the serial I/O input/output microprogram to gate data and controls to the card as follows:

'-bus out 0'

Data on '+com xmit data' or '+SIO xmit data' line.

'-bus out 3'

Allows '+com request to send' or '+SIO request to send' to be set or reset.

'-bus out 4'

Sets '+com request to send' or '+SIO request to send' when active and resets them when inactive.

'-bus out 5'

Allows rate data to be loaded to the card from the serial I/O input/output microprogram or communications I/O microprogram.

'-bus out 6'

Rate data (specifies the data rate from 20 to 9,600.5 bps).

'-bus out 7'

Not used by the expansion feature card.

Start Execute

The '+start execute' line is made active by the communications I/O microprogram or the serial I/O input/output microprogram to latch data and status information that is stored on the card to the communications I/O or serial I/O input/output microprogram. Data and status is sent on the bus in lines as follows:

'+bus in 1'

'+com clear to send' or '+SIO clear to send' (refer to *Communications Data Set EIA Interface* or *Serial I/O Device EIA Interface*).

'+bus in 2'

'+com data set rdy' or '+SIO data set rdy' (refer to *Communications Data Set EIA Interface* or *Serial I/O Device EIA Interface*).

'bus in 4'

'lg sp irpt' Defines the source of the program level 1 interrupt.

'bus in 7'

'+com rcvd data' or '+SIO rcvd data' (refer to *Communications Data Set EIA Interface* or *Serial I/O Device EIA Interface*).

Expansion Feature Card Transmit Operation

When the expansion feature card and the data set or I/O device are ready to transmit, the card interrupt logic and data rate clock are enabled. When an interrupt occurs at a preselected rate (20 to 9,600.5 bps as specified by the user), the communications I/O microprogram or serial I/O input/output microprogram sends the next bit to be transmitted to the card via '+bus out 0'; addresses the card (DA8); and activates the '-put strobe' line to gate the data bit to the card. The transmit operation occurs in the following sequence:

1. The fall of the data rate clock pulse on '-ungated osc' sets the timer interrupt latch. The timer interrupt latch sets '-irpt req' (interrupt request), which tells the communications I/O microprogram or serial I/O input/output microprogram that the card is ready to transmit a bit of data.

2. The data bit to be transmitted is sent to the card on '+bus out 0'.

3. The data bit and the rise of the data rate clock pulse on '+ungated osc' sets the data bit in the transmit buffer.

4. The bit is passed, via a driver circuit, to the '+com xmit data' line going to the data set or to the '+SIO xmit data' line going to the serial I/O device.

5. The timer interrupt latch is reset by the communications I/O microprogram or serial I/O input/output microprogram, and the sequence begins again to transmit the next bit.

The sequence continues until all bits for the character are transmitted. After transmitting the last bit, the communications I/O microprogram or serial I/O input/output microprogram disables the timer interrupt, but allows the clock to run, until the program is ready to send another character.

Expansion Feature Card Receive Operation

When the 5100 Portable Computer is in the receive state, the '+com rcvd data' or '+SIO rcvd data' line is monitored by the expansion feature card for a start bit (space level). When a start bit is detected, the data rate clock is started and a program level 1 interrupt is generated. (Refer to the theory section of the *IBM 5100 Maintenance Information Manual* for a description of a program level 1 interrupt.) This interrupt tells the communications I/O microprogram or serial I/O input/output microprogram that a bit has been received.

The microprogram in control addresses the card and activates the '-start execute bit' line (which samples the bit on the '+rcvd data' line instantaneously) to gate the bit to the communications I/O microprogram or serial I/O input/output microprogram.

Line noise only affects the value of the bit if it occurs when the bit is sampled. For example, if a noise pulse occurred at the exact same time as the '-start execute bit' line is activated (data bit is sampled), the noise pulse could be detected as a bit. However, because this is an extra bit, it could cause an error. (See *Error Detection*.)

The receive operation occurs in the following sequence:

1. The communications I/O microprogram or serial I/O input/output microprogram enables the start bit error detect logic. This logic monitors the '+com rcvd data' or '+SIO rcvd data' line for a space signal equal to at least one-half the bit time of the data rate being used.

2. During communications operations, the data set receives the start bit from the communications line and sends the bit on the '+com rcvd data' line to the 5100. The serial I/O device sends the start bit on the '+SIO rcvd data' line.

3. The '+com rcvd data' or '+SIO rcvd data' line is gated to 'bus in 7' and clock circuits. The start bit of each character received starts the data rate clock that runs until the complete character is received. Then the communications I/O microprogram or serial I/O input/output microprogram stops the clock and enables the start bit error detect logic ('+bus out 7') again.

4. The fall of the data rate clock pulse on '-ungated osc' sets the timer interrupt latch and, during serial I/O operation, latches the received data in the 'SIO data buffer'.

5. The microprogram recognizes the interrupt, reads the status latches, and resets the timer interrupt latch.

6. The sequence of steps 4 through 6 continues until the microprogram determines that all bits of the character are received.

7. After all bits for the character are received, the clock is turned off and the line is monitored for the next start bit.

Communications Transmit Interrupt

When the 5100 is transmitting, the remote system can interrupt the transmission by sending a long space (communications line is held at a space level (+) for a specified time). The '+com rcvd data' line is monitored during transmit for a continuous space greater than 63 ms. When a space level greater than 63 ms is detected, and '+com data set rdy' and '+com data term rdy' are active, a program level 1 interrupt is generated by the expansion feature card. The long space interrupt sets the long space interrupt status on '+bus in 4' to tell the communications I/O microprogram that this is a transmit interrupt. If a long space is detected while the 5100 is receiving, an interrupt is generated as is transmit, but it is ignored by the communications I/O microprogram.

Receive Interrupt

If the ATTN key on the 5100 is pressed during a communications receive operation, the communications I/O microprogram causes the adapter to transmit a space level on the '+com xmit data' line equivalent to three characters long. If the user selected EBCD and sends the character ⊞, the serial I/O input/output microprogram transmits a space level (three characters long) on the '+SIO xmit data' line.

ERROR DETECTION

The types of error detection performed by the 5100 during communications or serial I/O operations are:

● Parity errors — The characters are checked for correct parity.

● Start bit errors (receive only) — The characters are checked for a good start bit.

● Stop bit errors (receive only) — The characters are checked for a good stop bit.

Parity Errors

The communications I/O microprogram or serial I/O input/output microprogram maintains the correct parity of each character the 5100 transmits, and checks for the correct parity of each character the 5100 receives. See *Data Format* for the parity of each code the 5100 uses during communications or serial I/O operations.

Parity errors are indicated by an OUT (O backspace U backspace T) character on the display screen during communications or serial I/O operations and the character — in the communications diagnostic buffer during communications operations.

Start Bit Errors

During receive operations, the received data line is monitored for a space bit (+voltage level). (When the 5100 or remote system is in transmit mode but is not actually transmitting data, the transmitted data line is held at a mark (−) voltage level. Therefore, if the receive data line changes to the space level, it indicates the start of received data.) The start bit error detect logic checks this first bit to determine if it is a good start bit. A good start bit is at least one-half bit time in duration. If the bit is a good start bit, the data rate clock is allowed to run until the complete character is received. If the bit is not a start bit (it could be noise) the clock stops and the line is monitored for the next start bit.

There is no indication of a start bit error. However, a missing or distorted start bit could cause a parity error, a stop bit error, or both.

Stop Bit Errors

The communications I/O or serial I/O input/output microprogram checks each character for a stop bit by checking for the correct number of bits in each character. The correct number of bits in each character depends on the code being used (5-, 6-, 7-, or 8-bit). For example, a 6-bit character contains 9 bits, including the start and stop bits. Therefore, the ninth bit must be a stop bit or a stop bit error will occur.

Stop bit errors are indicated by an OUT character on the display screen during communications or serial I/O operations and an underscore (≐) character in the communications diagnostic buffer during communications operations.

Much of the hardware and many of the microprograms used during communications and serial I/O are also used in APL or BASIC language operations. You should run all the hardware diagnostic programs for better isolation when troubleshooting a communications or serial I/O problem. These diagnostic programs are described in the *IBM 5100 Maintenance Information Manual.*

Wrap connectors (P/N 1608894 and P/N 1608382) are required for the MDI (maintenance diagnostic integrated) program MDI 820. The wrap connectors can be attached to the end of the cables or to the 5100 communications or serial I/O connector. The wrap connectors connect the line request to send to clear to send, the line data terminal ready to data set ready, and transmitted data to received data. This allows the communications/serial I/O diagnostic MDI 820 to check the control and data lines with the 5100 Portable Computer isolated from the data set or serial I/O device. If the wrap connector is attached to the end of the data set or serial I/O adapter cable, the cable can also be tested. Refer to the *IBM 5100 Maintenance Information Manual* for the internal wiring of the wrap connectors.

COMMUNICATIONS DIAGNOSTIC BUFFER

All characters transmitted or received during communications are placed in the communications diagnostic buffer. This buffer is a 256-byte buffer in read/write storage; it is used to store all transmit and receive data. All characters (including control and function characters that are not normally displayed on the screen) that are received from or transmitted to the communications line are stored in the buffer. Therefore, the buffer can be used to isolate problems between the 5100 and the communications network.

Because the diagnostic buffer can contain only 256 characters, you might find it necessary to stop the communications operation to check the data desired.

You can stop the communications operation when you are transmitting data from the 5100 keyboard as follows:

1. Enter the data from the keyboard and press the EXECUTE key.

2. Press and hold the shift key.

3. Press the ATTN key.

4. Display the diagnostic buffer. (See *Displaying the Diagnostic Buffer.*)

You can stop the communications operation when the 5100 is transmitting data from tape as follows:

1. Press and hold the shift key.

2. Press the ATTN key.

3. Release the shift key and press the ATTN key again.

The tape operation stops and you can display the data in the buffer. (See *Displaying the Diagnostic Buffer.*)

You can stop the communications operation when the 5100 is receiving data as follows:

1. Press and hold the CMD key.

2. Press the ATTN key. (This procedure will cause the rest of the message to be lost and you might have to sign on or dial the remote system again to resume the receive operation.)

The receive operation will stop and you can display the last 256 bytes of data received in the buffer. (See *Displaying the Diagnostic Buffer.*)

Displaying the Diagnostic Buffer

To display the diagnostic buffer, enter the command &DIAG 0 (zero) as follows:

1. Press and hold the CMD key and press the 1 key. The character & is displayed on the screen.

2. Enter DIAG space 0 (zero) and press EXECUTE.

Refer to *CE Display* while reading the following description of how the data is displayed.

When the buffer is displayed:

• The data in the buffer is displayed in four lines at the top of the display screen.

• The buffer contains 256 characters; the most recent characters received or transmitted appear on the last line (line 4). The last character entered is displayed in the last position of the last line. When the buffer is full, the oldest character (first character on line 1) is lost when a new character is entered.

• Transmitted characters are underlined. An example of transmitted characters is shown on lines 1 and 3.

• Received characters are not underlined. An example of received characters is shown on lines 2 and 4.

• Function and control characters are displayed as special graphic characters. An example of function and control characters is shown on line 1.

• All data characters (BASIC or APL) are displayed as APL lower shift characters.

• Upper shift data characters are also displayed as APL lower shift characters preceded by an upper shift function character (>). An example of this is shown on lines 2 and 3.

Refer to *Communications Character Code Charts* in Appendix, *Character Code Charts,* to decode the characters in the buffer.

To determine what translation of the EBCD or correspondence code characters (APL or BASIC) the 5100 is using, search the extended display for the last &SYSTEM command entered, or display address hex 0142. (Refer to the *IBM 5100 Maintenance Information Manual, DCP1 Functions* for instructions on how to display data in read/write storage.) Check the &SYSTEM status bit (bit 15) of the displayed data, which indicates:

Bit 15	Translation
1	APL or APL.SV
0	BASIC or OTHER

To determine if the displayed data is upper shift or lower shift, scan right to left to the first shift function character. For example, in line 4 of the CE display, an upper shift function character (>) precedes the character A; therefore, the characters A through 0 are upper shift characters. If a (D) ( ] with EBCD and 9 with correspondence code) is encountered before a shift function character when scanning right to left, the characters to the right of the (D) are lower shift.

Note: The] or 9 can be either a data character or a (D) ; the (D) is normally preceded by a (C) (=).

3-2

The following example of data in the diagnostic buffer
(*CE Display*) shows how data, function, and control char-
acters appear in the communications diagnostic buffer. The
example of data in the extended display shows how the same
data appears in the extended display and on the display
screen.

CE Display

Lower Shift

Oldest   CE DISPLAY   Backspace   New-line
Character   Ⓒ   6 Spaces   Ⓓ 3

Line 1 — GHIJKLMNOPQRSTTUVWXYZ.[,]←×+/1234567890<K≤⊝≡]   →≡]AÑ<F≤⊝≡]AB

Line 2 — CDEFGHIJKLMNOPQRSTTUVWXYZ.[,]←×+/1234567890⊝→→→→   →≡]BÑ<F≤←≤←1
                              Upper Shift Data

Line 3 — KABCDEFGHIJKKLMNOPQRSTUVWXYZ.[,]←×+/1234567890K≤⊝≡]   →≡]BÑ<F

Line 4 — ≤⊝≡]<ABCDEFGHIJKLMNOPQRSTUVWXYZ.[,]←×+/1234567890≤⊝→→→→   →≡]←2
          Upper Shift   DEL Characters Used
          Function Character   for Timing Delay

[1] Upper shift.
[2] Last character entered.
[3] The Ⓓ is a] with EBCD. (See *Communications Code Charts.*)

Extended Display

Oldest Character in   Transmitted APL
CE Display   Lower Shift Characters

Line 1 ———————A←'ABCDEFGHIJKLMNOPQRSTTUVWXYZ.[,]←×+/1234567890'
Line 2 ——Transmitted ——→A   Received APL
Line 3 ————— ABCDEFGHIJKLMNOPQRSTTUVWXYZ.[,]←×+/1234567890←   Lower Shift
Line 4 ———————————B←'αι∩L∈_∇Δι°''□|⊤○*?ρ⌈∼↓∪ω⊃↑⊂:(;)→÷–\¨˜<≤≡≥>≠∨∧'
Line 5 ——Transmitted ——→B   Transmitted APL Upper Shift
Line 6 ————— αι∩L∈_∇Δι°'□|⊤○*?ρ⌈∼↓∪ω⊃↑⊂:(;)→÷–\¨˜<≤≡≥>≠∨∧
                         Received APL Upper Shift
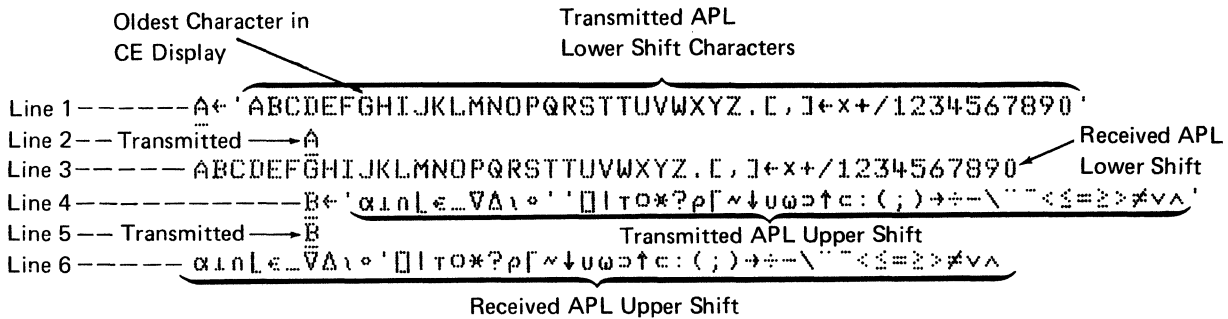
Line 1 of the CE display shows the transmitted message
(transmitted characters are underlined) that is shown on
line 1 of the extended display example. The first portion
of this message (A ← 'ABCDEF) has overflowed the buffer
and is not displayed on the CE display. The end of this
message is indicated by the first Ⓒ (=) on line 1.

Also on line 1 of the CE display is the message A that
was transmitted by the 5100 as shown on line 2 of the
extended display example. In this message you can see
the Ⓓ (]), which indicates the start of the message, and

the Ⓒ (=), which indicates the end of the message. The
rest of the characters between the Ⓓ and the Ⓒ make
up the message A.

The message A̲ was not transmitted as one character. This message is composed of the data and function characters An̄<F ≤ ⊙ as shown on line 1 of the CE display and was transmitted from the keyboard as follows:

1. The character A was entered and transmitted.

2. The backspace key was pressed, which caused the backspace function character n̄ to be transmitted.

3. The shift key was pressed and held, and the character __ (F key) was entered. This caused the uppershift function character < and the data character F to be transmitted.

4. The shift key was released and the EXECUTE key was pressed to indicate the end of the message. This caused the lower shift function character ≤ , the new-line character ⊙ , and the Ⓒ (=) to be transmitted.

The characters on the first part of line 2 of the CE display are lower shift characters that were received from the remote system. This message is shown on line 3 of the extended display example. To determine (in the CE display) that these are lower shift characters, scan right to left to the first shift function character or Ⓓ . In this message, a Ⓓ is encountered before a shift function character; therefore, the characters are lower shift.

The message on line 3 of the CE display contains transmitted upper shift characters because they are preceded by an upper shift function character (<) (the last character on line 2). This message is shown on line 4 of the extended display example.

NETWORK ANALYSIS SUBROUTINE

The network analysis subroutine measures the fidelity of the communications network. This provides the service personnel with a technique for isolating problems between the IBM 5100 Portable Computer and the communications network.

The network analysis subroutine samples and analyzes an alternating mark and space pattern of either 134.5 or 300 bps received by the 5100. Samples are taken of the '+rcvd data' line at a uniform rate of five samples per bit (Figures 1 and 2). Any samples in error are accumulated in counters and are displayed at the end of the test. (See *How Data Is Accumulated*.) The total number of bits sampled is also counted. Sampling is synchronized for every bit by a hardware timer interrupt that ideally occurs at the midpoint of the bit. The hardware timer is synchronized every 20 bits by detecting and locking on the midpoint of a start (space) bit.

With the start bit detect circuitry enabled, an interrupt occurs halfway through the next space bit if there is no noise on the bit. If the signal is noisy and the interrupt does not occur within the allotted time period, a sync error is detected and stored in a counter and the next space bit is used as a sync bit. When 256 consecutive sync errors occur, the sampling ceases and an error message is displayed.

When the signal is good and the interrupt for the start bit occurs, the next five samples are checked to determine if the interrupt was approximately at the midpoint of the start bit. If all five samples are good, the next 19 bits are sampled and the samples analyzed before resynchronizing the hardware timer. If one or more of the first five samples are bad, a sync error is accumulated; and the center sample of the bit plus all other samples detected as bad are accumulated in the error counters. As the circuitry is set to detect the next start bit, the intervening mark bit is not sampled.
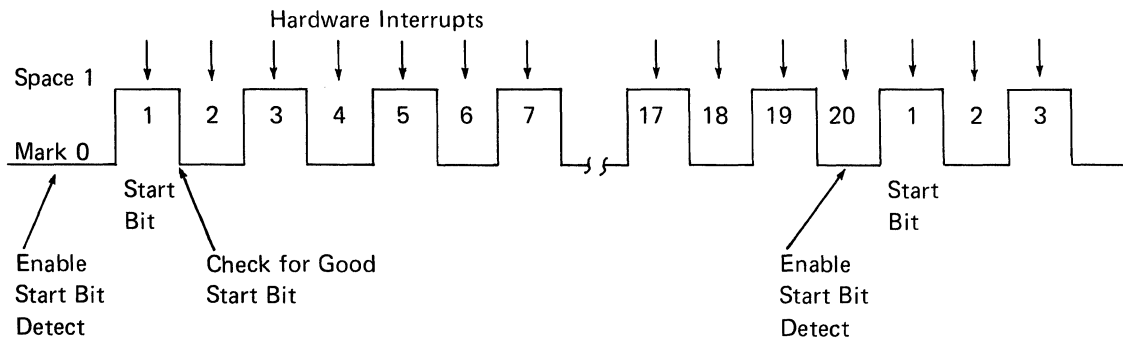
Figure 1. Synchronization Sequence
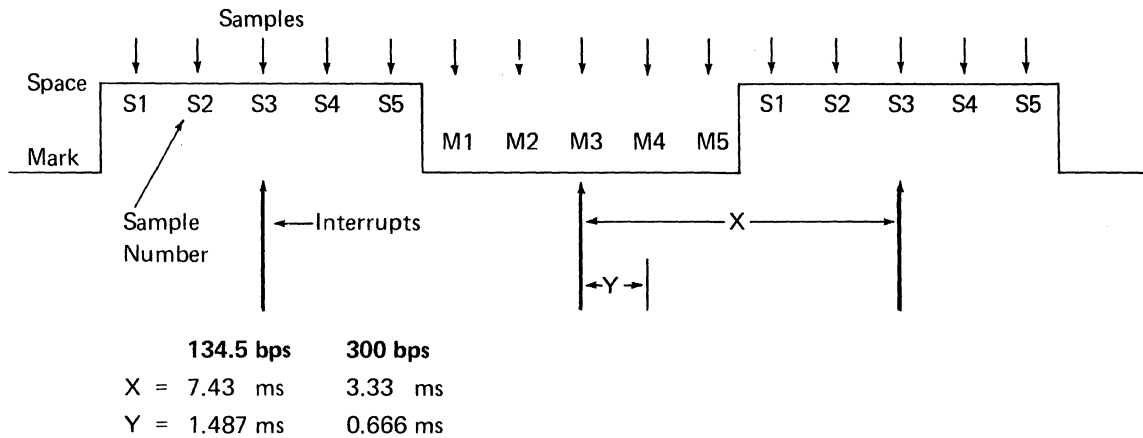
Figure 2. Sample Sequence

	134.5 bps	300 bps
X =	7.43 ms	3.33 ms
Y =	1.487 ms	0.666 ms

Operating Instructions

To operate the network analysis subroutine:

1. Load DCP2 (diagnostic control program 2) and select MDI (maintenance and diagnostics integrated) 894 with the options specified.

2. Press EXECUTE to load the selected MDI and to display the MDI options.

3. Select the step mode and enter step 31 for 134.5 bps or step 32 for 300 bps.

4. Press EXECUTE. The display then indicates the procedure for establishing a data link with the facility that will transmit the alternating test pattern.

5. Dial the remote facility and establish a voice link.

6. Coordinate the type of test to be performed and the procedure to be used to establish the data link (channel).

7. Press EXECUTE to make the IBM 5100 ready and initiate the subroutine.

8. Place the answering data set in the data mode by pressing the DATA key.

9. When the carrier tone is heard in the originating data set, place it in data mode by pressing the DATA key on that data set.

The status displayed at the bottom of the display screen should proceed from DS NOT READY when EXECUTE is first pressed to NOT CLR TO SEND when the DATA key is pressed, then to RECEIVING when the channel has been established and data is being exchanged. If DS NOT READY status remains after the DATA key is pressed, check that the 5100 is connected to the telephone line. If NOT CLR TO SEND status remains after the DATA key has been illuminated for approximately 5 seconds, press the TALK key, reestablish the voice link, and try again.

To return to the voice mode from the data mode, lift the handset and press the TALK key. The tone should cease and the voice link should be reestablished when the TALK key is pressed on the remote data set.

Pressing ATTN while DS NOT READY or NOT CLR TO SEND status is indicated terminates the test.

The following paragraphs describe the status messages and provide possible causes of the DS NOT READY and NOT CLR TO SEND messages:

DS NOT READY — Ready status was not received from the data set because:

1. The data set is not connected to the telephone line,

2. The data set is not in the data mode,

3. The data set cable is not connected to the 5100,

4. The 5100 is not generating the 'data term rdy' signal properly, or

5. The data set is not operating properly.

NOT CLR TO SEND – The data set is ready, but 'clear to send' was not received from the data set because:

1.   The carrier from the distant data set is not detected,

2.   The 5100 is not providing the 'request to send' signal properly, or

3.   The data set is not operating properly.

RECEIVING – Data link is established.

When the network analysis subroutine is executed, it:

● Initializes registers and error counters.

● Performs the data link handshaking functions and displays the status message RECEIVING.

● Synchronizes sampling to a good start bit.

● Samples the '+rcvd data' line and analyzes the samples.

● Synchronizes sampling every 20 bits.

Sampling is terminated when ATTN is pressed, when one of the sample error counters reaches its maximum value of 64K, or when an error condition is detected. If ATTN is pressed or if one of the counters becomes full, the first graph is displayed (see *Displays*). If an error is detected, the appropriate error message is displayed. The error messages are:

● INITIAL SYNC ERROR   – 512 bit times elapse without establishing synchronization.

● RESYNC ERROR   – 256 consecutive bit times elapse without reestablishing synchronization.

● INTERRUPT ERROR   – Hardware interrupt fails between sync times.

● DATA LINK ERROR   – Data link is disconnected due to loss of either 'data set rdy' or 'clear to send' signal or both.

After viewing the error message and/or each page of results, four options are available:

1.   Page to the next graph. Paging from the last graph returns to the first graph.

2.   Continue the test with all the error counters unaltered.

3.   Restart the test with all the error counters reinitialized.

4.   Terminate the test.

Remote End Operation

The network analysis subroutine requires a continuous alternating 1/0 pattern on the '+rcvd data' line. The bit rate may be either 134.5 or 300 bps. The rate is set by the entry used to start the MDI routine.

The source of the alternating pattern may be:

● Remote system software routine

● CE test panel

● Teleprocessing diagnostic analyzer tester (TDAT)

When using remote system software routines, these routines must provide a continuous alternating pattern. This pattern must not be modified by the communication line interface through the insertion of formatting bits or control characters. CE test panel on the 370X communication controllers can be connected to a communication line and transmit the required pattern. For the correct procedure, refer to the maintenance manual for the controller being used.

An alternate procedure that can be used involves the use of the TDAT. With this procedure, the 5100 transmit routine provides an alternating pattern that is recorded on tape by the TDAT. The recorded data is then played back to the 5100 by the TDAT for the network analysis subroutine. Refer to the *1200 Teleprocessing Diagnostic Analyzer Tester* manual, S226-3029, for this procedure.

How Data Is Accumulated

Counters accumulate the following information:

1. Total number of bits sampled

2. Number of sync errors

3. Number of sample errors for each sample position for both mark and space bits (Figure 2)

4. Number of consecutive error-free bits

5. The number of error-free bit strings in lengths ranging from less than 16 bits to greater than 64K bits

After each bit is sampled five times, each sample is checked for correctness by comparing its value to the expected value for that sample. Any incorrect sample increments the corresponding sample error counter.

The center-bit sample is used to determine consecutive error-free bit counts. The consecutive error-free bit counter is incremented after every consecutive good center-bit sample. Incrementing of the counter ceases when it reaches its maximum value of 64K. When the center-bit sample is incorrect, the magnitude of the consecutive error-free bit counter is checked to determine which error-free bit string counter to increment. The consecutive error-free bit counter is then reset to 0 to prepare for the next error-free bit string. Sync errors are treated the same as center-bit errors for error analysis purposes.

Displays

Results of the network analysis subroutine are displayed in the form of three bar graphs on the display screen:

1. Space errors by sample

2. Mark errors by sample

3. Error-free bit strings

Space and Mark Error Graphs

The space and mark error graphs each consist of five horizontal bars of asterisks that plot the number of incorrect levels discovered on the '+rcvd data' line at each of the five uniform bit sampling times. The actual number of errors for each sample time is also displayed as a 5-digit decimal number in the total column at the extreme right of the graph.

The scale of errors along the bottom of these graphs is not linear. The value of each space on the graph increases with each 10-space division from left to right. For example, on the extreme left of the scale, in the 0-16 error range, every two asterisks in a bar represent approximately three errors. In the middle of the scale, between the 64 and 256 divisions, each asterisk represents approximately 19 errors. At the extreme right of the scale, between 512 and 1K divisions, each asterisk represents approximately 50 errors.

The total associated with each sample displays the actual number of errors for that sample. Although the graph only goes to 1K, up to 64K can be displayed in the total column for each sample.

Displayed on the bottom line of these graphs are the total number of bits received during the test, the number of bit synchronization errors, and the data transmission rate being tested (either 134.5 or 300 bps).

```
MDI 894          STEP NO. 031    -->
                    SPACE ERRORS BY SAMPLE


SAMPLE                                                                  TOTAL
  1       |                                                            00000
  2       |****                                                        00008
  3       |**************************************                      00287
  4       |*********                                                   00016
  5       |*******************                                         00068
                 |          |          |          |          |
ERRORS:          16         64         256        512        1K
         TOTAL BITS=04369255    SYNC ERRORS=00170    RATE=134.5 BPS

ENTER P(PAGE), C(CONTINUE), R(RESTART) OR T(TERMINATE) P
STATUS: ENTER REPLY


MDI 894          STEP NO. 032    -->
                    MARK ERRORS BY SAMPLE


SAMPLE                                                                  TOTAL
  1       |**********************************************              00512
  2       |**********************************************              00568
  3       |*****************************************************       01024
  4       |******************************************              00341
  5       |***********************************************************  65535
                 |          |          |          |          |
ERRORS:          16         64         256        512        1K
         TOTAL BITS=04369255    SYNC ERRORS=00170    RATE= 300 BPS

ENTER P(PAGE), C(CONTINUE), R(RESTART) OR T(TERMINATE) P
STATUS: ENTER REPLY
```

Error-Free Bit Strings

The third graph plots the occurrences of various lengths of
error-free bit strings received during the test. An error-free
bit string is a consecutive series of bits in which the center
sample (sample 3, see Figure 1) was correct. The beginning
of an error-free bit string is either the start of the test or the
first correct bit after the last error bit. The end of an error-
free bit string is the bit preceding an error bit. The error-
free bit string is a measurement of the time that the com-
munications network operated error-free between failures.

Bit string lengths are divided into eight ranges. Each occur-
rence of an error-free string of bits causes the counter for
the proper length range to be incremented when the string
is ended by an error.

Plotted on the graph is the value of each of the eight length
range counters as a horizontal bar of asterisks. Each counter
value is also displayed as a 5-digit decimal number in the
total column at the extreme right of the graph. As with the
space and mark sample graphs, up to 64K can be displayed
in the total column for each error-free range.

The scale of occurrences along the bottom of the graph is
not linear. The value of each space on the graph increases
with each 10-space division from left to right. For example,
on the extreme left of the scale, every two asterisks in a bar
represent approximately three occurrences of a bit string.
In the middle of the scale, between the 64 and 256 divisions,
each asterisk represents approximately 19 occurrences of a
bit string. At the extreme right of the scale, between the
512 and 1K divisions, each asterisk represents approximately
50 occurrences of a bit string.

Also displayed on the bottom line of the graph are the
total number of bits received during the test and the current
value of the error-free bit counter. The error-free bit
counter indicates the number of bits received by the net-
work analysis subroutine since the last bit error or since
the beginning of the test if no bit errors were detected.

```
MDI 894          STEP NO. 031    ->
RANGE (BITS)              ERROR-FREE BIT STRINGS                                    TOTAL
  >64K   | *                                                                       00002
16K-64K | ***********                                                              00020
 4K-16K | *********************                                                    00090
  1K-4K | ****************************                                             00128
 256-1K | ****************                                                         00040
 64-255 | ************************                                                 00089
 16-63  | **************                                                           00040
  0-15  | ************************************************************************  01024
          |          |          |          |          |
OCCURRENCES:  16        64        256        512         1K
       TOTAL BITS=04369255      CONSEC ERROR-FREE BITS=65535


ENTER P(PAGE), C(CONTINUE), R(RESTART) OR T(TERMINATE) P
STATUS: ENTER REPLY
```

```
    ( Restart )              ( Start )              ( Continue )
        │                       │                       │
        │                       │                       │
  ┌─────┴─────┐           ┌─────┴─────┐           ┌─────┴─────┐
  │ Restart entry │        │ Set timer and │        │ Continue entry │
  │ from options  │        │ sample rates  │        │ from options   │
  └─────┬─────┘           └─────┬─────┘           └─────┬─────┘
        │                       │                       │
        └───────────────────────┤                       │
                                │                 ┌─────┴─────┐
                          ┌─────┴─────┐           │ Restore consecutive │
                          │ Initialize register │  │ error-free bit count │
                          │ and error counters  │  └─────┬─────┘
                          └─────┬─────┘                 │
                                │◄────────────────────────┘
                          ┌─────┴─────┐
                          │ Data link │
                          │ handshaking │
                          └─────┬─────┘
            ┌──────────────────►│
            │             ┌─────┴─────┐
            │             │ Set up to detect │
            │             │ start bit │
            │             └─────┬─────┘
            │             ┌─────┴─────┐
            │             │ Wait for interrupt │
            │             └─────┬─────┘
            │                   │
            │                  ╱ ╲                    ┌─────────────┐
            │                 ╱   ╲      Yes          │ Sample start bit │
            │                ╱Interrupt╲──────────────┤             │
            │                ╲         ╱              └─────┬───────┘
            │                 ╲       ╱                     │
            │                  ╲     ╱                      │
            │                   │ No                        │
            │                   │◄──────────────┐          │
            │                  ╱ ╲               │         ╱ ╲
       No   │                 ╱   ╲              │        ╱   ╲      Yes
   ◄────────┼────────────────╱256 sync╲          │       ╱Good start╲─────┐
            │                ╲ errors ╱          │       ╲   bit   ╱      │
            │                 ╲      ╱           │        ╲       ╱      ▼
            │                  ╲    ╱            │         ╲     ╱      ╱1╲
            │                   │ Yes            │          │ No       ▽
            │             ┌─────┴─────┐          │    ┌─────┴─────┐
            │             │ Display initial sync │    │ Reset interrupt │
            │             │ error message │       │    └─────┬─────┘
            │             └─────┬─────┘          │          │
            │                   │                └──────────┘
            │             ┌─────┴─────┐
            │             │ Exit to options │
            │             └───────────┘
```

3-10

```
                    ┌───┐
                    │ 1 │
                    └─┬─┘
                      │
        ┌─────────────────────────┐
        │ Increment total         │
        │ bit counter             │
        └─────────────┬───────────┘
                      │
        ┌─────────────────────────┐
        │ Increment consecutive   │
        │ error-free bit counter  │
        └─────────────┬───────────┘
                      │
        ┌─────────────────────────┐
        │ Reinitialize for        │
        │ analysis                │
        └─────────────┬───────────┘
                      │
        ┌─────────────────────────┐
        │ Wait for interrupt      │
        │                         │
        └─────────────┬───────────┘
                      │
                   ◇ Interrupt ◇ ──── No ────┐
                      │                        │
                    Yes                        │
        ┌─────────────────────────┐   ┌─────────────────────────┐
        │ Sample bit              │   │ Display interrupt       │
        │                         │   │ error message           │
        └─────────────┬───────────┘   └─────────────┬───────────┘
                      │                              │
        ┌─────────────────────────┐        ( Exit to options )
        │ Analyze samples         │
        │                         │
        └─────────────┬───────────┘
                      │
        ┌─────────────────────────┐
        │ Increment total         │
        │ bit counter             │
        └─────────────┬───────────┘
                      │
          No ──── ◇ 19th bit ◇
                      │
                    Yes
                    ┌───┐
                    │ 2 │
                    └───┘
```

```
                                    ┌──┐
                                    │ 2│
                                    └──┘
                                      │
                                      ▼
                          ┌───────────────────────┐
                          │ Set up to detect       │
                          │ start bit              │
                          └───────────────────────┘
                                      │
                                      ▼
                          ┌───────────────────────┐
                          │ Wait for interrupt     │
                          └───────────────────────┘
                                      │
                                      ▼
    ┌───────────────────┐           ╱╲
    │ Sample start bit   │◄──Yes──╱      ╲
    └───────────────────┘        ╱ Interrupt ╲
             │                    ╲          ╱
             ▼                      ╲      ╱
           ╱╲                         ╲╱
         ╱    ╲                        │ No
  Yes  ╱        ╲                      ▼
◄─────╱ Good start ╲        ┌───────────────────────┐
      ╲    bit    ╱         │ Increment resync       │
   ┌──┐ ╲        ╱          │ error counter          │
   │ 1│   ╲    ╱            └───────────────────────┘
   └──┘     ╲╱                        │
             │ No                     ▼
             ▼            ┌───────────────────────┐
    ┌───────────────────┐ │ Set center bit         │
    │ Reset interrupt    │ │ sample error           │
    └───────────────────┘ └───────────────────────┘
                                      │
                                      ▼
                          ┌───────────────────────┐
                          │ Analyze samples        │
                          └───────────────────────┘
                                      │
                                      ▼
                          ┌───────────────────────┐
                          │ Increment total bit    │
                          │ counter                │
                          └───────────────────────┘
                                      │
                                      ▼
  ┌───────────────────┐             ╱╲
  │ Display resync     │           ╱    ╲
  │ error message      │◄──Yes──╱   256    ╲──No──►
  └───────────────────┘        ╲ consecutive ╱
           │                    ╲  resync   ╱
           ▼                      ╲ errors ╱
    ╭───────────────╮               ╲╱
    │ Exit to options │
    ╰───────────────╯
```

COMMUNICATIONS/SERIAL I/O
TROUBLESHOOTING GUIDE

Use the following chart as a guide when troubleshooting communications or serial I/O problems:

- Run MDI 820; the communications/serial I/O diagnostic MDI 820 tests the communications and serial I/O hardware (B2 card, internal cables, and external cables). If you suspect the internal cables (cables from Z1 socket to the serial I/O communications connector — see the *IBM Maintenance Information Manual*), you can jumper the following signals at the Z1 socket and run MDI 820:

> Data terminal ready to data set ready
> Request to send to clear to send
> Transmitted data to received data

> If the problem is intermittent, run MDI 820 several times.

DID MDI 820 RUN ERROR FREE?
Y N
| — Correct the problem indicated by MDI
| 820 and rerun MDI 820. If the diagnostic
| runs error free, retry the user's job.
|
| DOES THE USER'S JOB STILL FAIL?
| Y N
| | — Good!
| |
| — Go to entry 002.

002

- Run the 5100 diagnostics. Because the communications and serial I/O features use existing ROS microprograms and microprograms in read/write storage, you should run the 5100 diagnostics to eliminate possible problems in these areas.

DID THE 5100 DIAGNOSTICS RUN ERROR
FREE?
Y N
| |
| |
| |
A B

A B

Y N
| — Correct the problem indicated by the
| 5100 diagnostics and rerun the user's
| job. If you are unable to correct the
| problem indicated by the 5100 diag-
| nostics, use existing support procedures.
|
| DOES THE USER'S JOB STILL FAIL?
| Y N
| | — Good!
| |
| — Go back to entry 002.

ARE YOU USING THE COMMUNICATIONS
FEATURE?
Y N
| — The 5100 appears to be functioning
| normally. Check to make sure that the
| correct serial I/O cable is connected to
| the I/O device and that the user speci-
| field the correct device characteristics.
| (See the *IBM 5100 Serial I/O Adapter
| Feature User's Manual*.)

IS THE 5100 USING SWITCHED LINES?
Y N
| — Use the 5100 communications diagnostic
| buffer to verify that the data is being
| transmitted and received by the 5100
| correctly. If the user's job still fails,
| have the user contact the telephone
| company and request a line test. If the
| job still fails, use existing support proce-
| dures.

— If you dial the remote system again (two or three times if necessary and wait 15 minutes between dialing), you might establish a different route on the communications network. This procedure could bypass possible network problems.

DOES THE USER'S JOB STILL FAIL?
Y N
| — If redialing corrected the problem, there
| are possible problems with the com-
| munications network. If the problem
| persists, have the user contact the tele-
| phone company.

C

C

Y
|
— Dial a system with test capability and send
  a test message. If this is not possible, dial
  the user's remote system and send a test
  message.

DID THE REMOTE SYSTEM RECEIVE THE
TEST MESSAGE CORRECTLY?
Y N
  | — Check the data in the 5100 communica-
  | tions diagnostic buffer.
  |
  | IS THE DATA CORRECT IN THE BUFFER?
  | Y N
  | | — The problem appears to be in the
  | | 5100. Use existing troubleshooting
  | | and support procedures.
  | |
  | — The 5100 appears to be transmitting cor-
  | rectly. Retry the user's job; if it still
  | fails, ask the user to contact the tele-
  | phone company and request a line test.
  | If the job still fails, use existing support
  | procedures.
  |
— Have the remote system system send a test
  message to the 5100.

DID THE 5100 RECEIVE THE TEST MES-
SAGE CORRECTLY?
Y N
  | — Use the 5100 communications diagnostic
  | buffer to check the received data.
  |
  | IS THE DATA IN THE BUFFER CORRECT?
  | Y N
  | | — The data was not received correctly.
  | | The remote system did not send the
  | | correct data, or there are possible
  | | problems with the communications
  | | network. Have the user contact the
  | | telephone company and request a
  | | line test. If the job still fails, use
  | | existing support procedures.
  | |
  | — The problem appears to be in the 5100.
  | Use existing troubleshooting and sup-
  | port procedures.
  |
— The 5100 appears to be receiving correctly.
  Retry the user's job; if it still fails, use
  existing support procedures.

A communications network consists of the equipment and
transmission lines that transmit data between terminals
and computers. The following diagram shows a typical
communications network involving the IBM 5100 and a
remote system. Because communications network config-
urations vary from network to network, this diagram may
not show the network configuration involving your 5100.
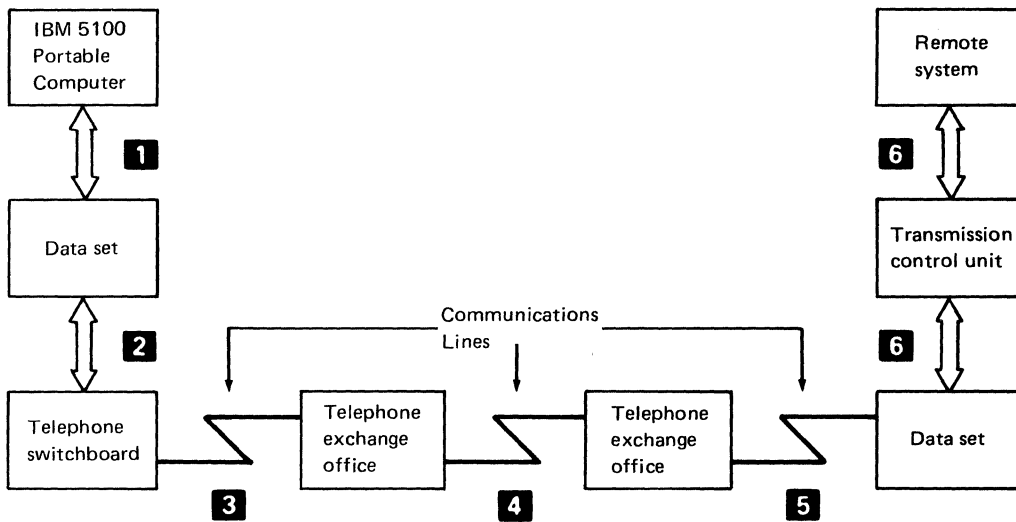
In this network, the data flow between the IBM 5100
Portable Computer and the remote system is as follows:

1 The 5100 sends the data to the data set.

2 The data set sends the data to a telephone
switchboard.

3 The telephone switchboard routes the data to the
local telephone exchange office via communications
lines.

4 The local telephone exchange office sends the data
to the telephone exchange office at the location of
the remote system via communications lines.

5 The telephone exchange office sends the data to the
data set at the location of the remote system.

6 The data set sends the data to the transmission control
unit and remote system.

SWITCHED NETWORKS

The communications network in the preceding example is called a switched network. A switched network uses the same lines and exchange equipment that telephone companies use for normal voice communications. To establish a connection between the 5100 and a remote system, you must dial the remote location.

Switched lines may take different routes each time you dial a certain location. Some reasons for this include situations, such as unavailable lines during peak calling periods, and problems with equipment. It is common for data being transmitted a few hundred miles to actually travel thousands of miles because of busy lines during peak calling hours. As it travels, the data can encounter many devices, such as amplifiers, microwave equipment, and terminating and switching equipment. Therefore, if you have transmission problems, dialing again might establish a different route, which could solve the problem.

NONSWITCHED NETWORKS

Unlike switched networks, nonswitched networks (leased lines) always use the same route to the remote system and do not require dialing to establish the data connection. As in switched networks, nonswitched lines may also be routed through many different network devices, such as amplifiers, exchange equipment, and microwave equipment. When you have problems with nonswitched lines, you cannot establish a different route by dialing. You will usually find it necessary to work with a representative from a telephone company.

COMMON TERMS

When you are working with a representative of a telephone company, it will be helpful if you are familiar with the terms that are used to describe problems and components of a communications network. The following are some of the terms that apply to the 5100 communications and serial I/O adapter features:

Asynchronous is a type of transmission in which each character is synchronized individually with a start and stop bit. (See Data Format in Theory of Operation.)

Baud is a term that represents the data rate. One baud is equal to 1 bit of data per second (1 bps). For example, 300 baud is equal to 300 bits per second.

Carrier is a high-frequency signal that is modulated by voice or data signals (the frequency depends on the data set design). When you dial the remote system, the tone you hear in the receiver is the carrier. If the tone is steady, it is not modulated. If the tone sounds like many tones combined (varying tone), it is modulated.

Data phone[1] is a telephone designed for data transmission.

Data set (modem) is a device that supplies the carrier during data communications, modulates the carrier with the data signals, and transmits it over the communications lines. The data set also receives the modulated carrier from the communications lines, demodulates it (removes the data from the carrier) and sends it to the terminal or computer.

Demodulation is the process of removing the data from the carrier.

Duplex indicates that data can be transmitted in both directions.

Full duplex indicates that data can be transmitted in both directions simultaneously.

Half duplex indicates that data can be transmitted in both directions, but not simultaneously.

Line adapter is an impedance matching device used to connect other devices to communications lines. Line adapter is also used, at times, as another term for data set or modem.

Line hit is electrical interference that causes spurious signals on a data circuit.

Mark indicates the presence of a bit on a data circuit. (See Data Format in Theory of Operation.)

Modem. (See data set under Common Terms.)

Off hook indicates that the line is connected or active.

On hook indicates that the line is disconnected or inactive.

_____
[1] A trademark of the Bell system.

Ring side is the conductor of a cable that is connected
to the ring of a plug or jack.

Example:

Ring ——————⊙—————— Tip
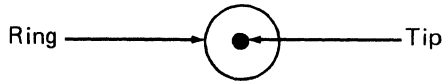
Space indicates no data bit on a data circuit.

Start-stop is a term for asynchronous data transmission
using an ST (start) and an SP (stop) bit to synchronize
each character on the data circuit. (See *Data Format*
in *Theory of Operation*.)

Tip side is the conductor that is connected to the center
or tip of a plug or jack. (See *ring side* under *Common
Terms*.)

DATA SET FUNCTIONS

The data set converts the data signals from a terminal or
system to signals that can be transmitted over the com-
munications network, and converts the signals from the
communications network to signals that can be used by
the terminal or system. The following diagram and brief
discussion show how this is done normally.
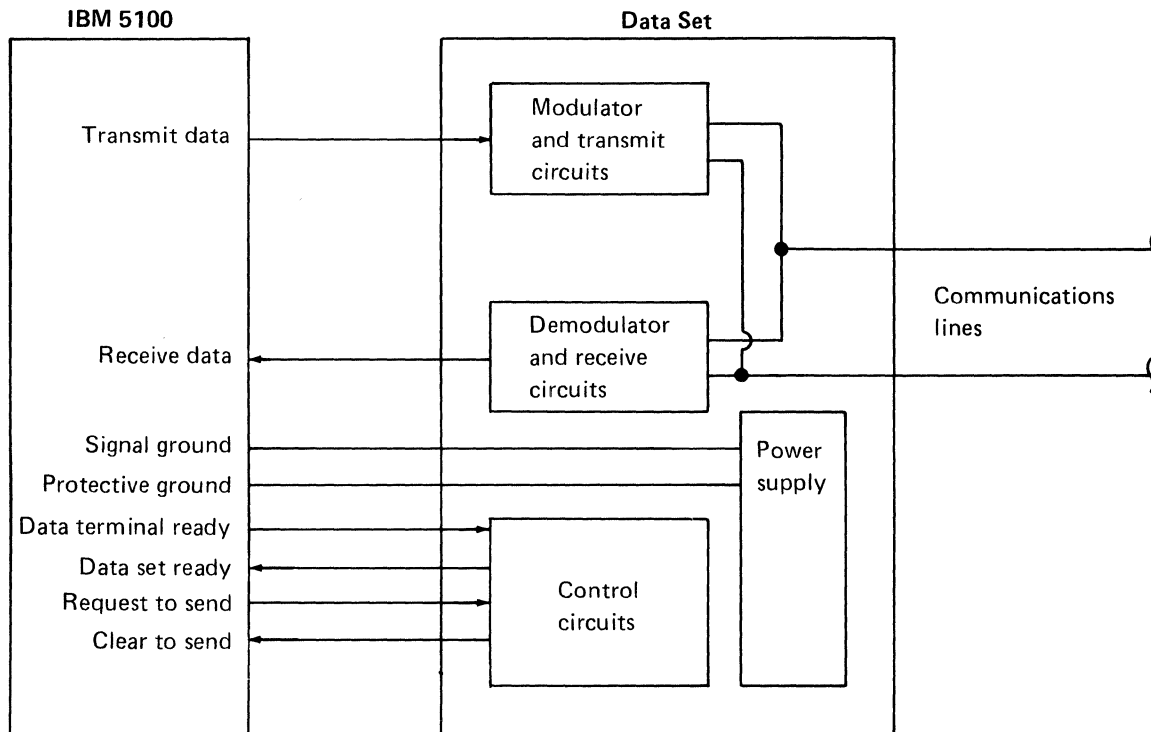
A data set has circuits to transmit a carrier on the communications lines, circuits to modulate the carrier with data signals during transmit operations, and circuits to demodulate the carrier during receive operations. The carrier is used to minimize the affects of noise and distortion on the data signals and allow transmission of more than one data signal on the same network simultaneously. In addition, there are control circuits to sense and activate signals that control the data set and the terminal or system communications operation. The control signal line names, pin numbers, and voltage levels are standardized by the EIA (Electronics Industry Association) and are called the EIA standard interface. (See *Communications Data Set EIA Interface* or *Serial I/O Device EIA Interface* for an explanation of the EIA signals used with the 5100.)

The data set sequence of operations when the 5100 is transmitting data is as follows:

1. The 5100 turns on data terminal ready to indicate that the 5100 is ready to communicate.

2. The data set turns on data set ready to indicate that the data set is ready to communicate.

3. The 5100 turns on request to send to indicate that the 5100 is ready to transmit data.

4. The data set turns on clear to send to indicate that the data set is ready to transmit data and turns on the carrier. When the 5100 is not sending data, the carrier is held at a mark frequency.

5. The 5100 sends the data on the transmit data line to the data set.

6. The data set modulates the carrier with the data signals and transmits the signals over the communications lines.

When the 5100 is receiving data:

1. Data terminal ready and data set ready are turned on as in a transmit operation.

2. The data set receives the data from the communications lines in the form of a modulated carrier. When no data is being transmitted, the carrier is held at a mark frequency.

3. The data signals are separated from the carrier by the demodulation circuits and sent to the 5100 on the received data line. The first bit of a character (ST bit) is a space level (+ voltage) on the received data line which indicates the start of a character to the 5100.

After all data is transmitted or received, the control lines (except data terminal ready and originate) and carrier are turned off until the 5100 requests another transmit or receive operation.

The following character code charts show the EBCD corre-
spondence code, 5-, 6-, and 7-bit characters, the corre-
sponding 5100 characters, and their bit values as they
appear on the receive data and transmit data lines. These
charts do not indicate the characters as they appear on the
5100 keyboard. The 8-bit code uses the hexadecimal values
00 through FF.

COMMUNICATIONS CODE CHARTS

The communications code charts show the EBCD and correspondence code data, function, and control characters that are used with the communications feature. Use these charts to determine the bit values of the characters on the transmit data and receive data lines and determine what the graphics indicate that appear in the diagnostic buffer.

Code Chart of Function and Control Characters (For Both EBCD and Correspondence Code)

Character in Communications Diagnostic Buffer	B	A	8	4	2	1	C	Meaning	
<			8	4	2			Upper shift	
=			8	4	2	1	C	Ⓒ	
©	B		8	4		1	C	New line	These characters can be received and transmitted.
n̄	B		8	4	2		C	Backspace	
⊂	B	A	8	4		1		Tab	
≤	B	A	8	4	2			Lower shift	
]			8		2	1		Ⓓ[1]	
Blank								Space	
↓			8	4			C	Punch on	
↑			8	4		1		Reader stop	
∧		A	8	4	2		C	EOB	These characters are received, but are ignored. They cannot be transmitted.
..		A	8	4	2	1		Prefix	
U	B	A	8	4			C	Punch off	
→	B	A	8	4	2	1	C	DEL	
\		A				1	C	Invalid idle	
—	B							Invalid idle	
÷	B	A					C	Invalid idle	
~	B		8	4	2	1		Valid idle	
ω		A	8	4				Bypass	These characters are received and used. They cannot be transmitted.
≠		A	8	4		1	C	Line feed	
⊃	B		8	4				Restore	
=								Receive parity error	
÷								Receive stop bit error	
\								Indicates a transmit long space	

[1] This character can also indicate a] or) data character in the diagnostic buffer when using EBCD or a 9 or ⌣ data character when using correspondence code.

Code Chart for EBCD Data Characters

| GRAPHIC CHARACTERS | | | | | | EBCD BIT VALUE | | | | | | |
APL LOWER SHIFT	APL UPPER SHIFT	BASIC LOWER SHIFT	BASIC UPPER SHIFT	EBCD LOWER SHIFT	EBCD UPPER SHIFT	B	A	8	4	2	1	C
A	α	A	A	a	A	B	A				1	
B	⊥	B	B	b	B	B	A			2		
C	∩	C	C	c	C	B	A			2	1	C
D	⌊	D	D	d	D	B	A		4			
E	∊	E	E	e	E	B	A		4		1	C
F	‾	F	F	f	F	B	A		4	2		C
G	∇	G	G	g	G	B	A		4	2	1	
H	∆	H	H	h	H	B	A	8				
I	ι	I	I	i	I	B	A	8			1	C
J	∘	J	J	j	J	B					1	C
K	·	K	K	k	K	B				2		C
L	▯	L	L	l	L	B				2	1	
M	|	M	M	m	M	B			4			C
N	⊤	N	N	n	N	B			4		1	
O	○	O	O	o	O	B			4	2		
P	*	P	P	p	P	B			4	2	1	C
Q	?	Q	Q	q	Q	B		8				C
R	ρ	R	R	r	R	B		8			1	
S	⌈	S	S	s	S		A			2		C
T	~	T	T	t	T		A			2	1	
U	↓	U	U	u	U		A		4			C
V	∪	V	V	v	V		A		4		1	
W	ω	W	W	w	W		A		4	2		
X	⊃	X	X	x	X		A		4	2	1	C
Y	↑	Y	Y	y	Y		A	8				C
Z	⊂	Z	Z	z	Z		A	8			1	
.	:	'	¬	.	¬	B	A	8		2	1	
[	(	$	!	$	!	B		8		2	1	C
,	;	,	|	,	|		A	8		2	1	C
]¹	)	#	··	#	··			8		2	1	
←	→	@	¢	@	¢		A					
×	÷	&	+	&	+	B	A					C
+	—	—	—	—	—	B						
/	\	/	?	/	?		A				1	C
1	··	1	=	1	=						1	
2	¨	2	<	2	<					2		
3	<	3	;	3	;					2	1	C
4	≤	4	:	4	:				4			
5	=	5	%	5	%				4		1	C
6	≥	6	'	6	'				4	2		C
7	>	7	>	7	>				4	2	1	
8	≠	8	*	8	*			8				
9	∨	9	(	9	(			8			1	C
0	∧	0	)	0	)			8		2		C

These characters appear in the communications diagnostic buffer.

[1] The character] can also indicate a Ⓓ in the diagnostic buffer.

Code Chart for Correspondence Code Data Characters

GRAPHIC CHARACTERS						CORRESPONDENCE BIT VALUE						
APL		BASIC		CORRESPONDENCE								
LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	B	A	8	4	2	1	C
A	α	A	A	a	A	B			4	2	1	C
B	⊥	B	B	b	B		A	8		2	1	C
C	∩	C	C	c	C		A		4	2	1	C
D	⌊	D	D	d	D		A		4		1	
E	∊	E	E	e	E		A		4			C
F	¯	F	F	f	F	B	A			2	1	C
G	∇	G	G	g	G	B	A				1	
H	∆	H	H	h	H		A	8			1	
I	ι	I	I	i	I	B			4	2		
J	∘	J	J	j	J	B	A					C
K	'	K	K	k	K		A		4	2		
L	□	L	L	l	L		A	8				C
M	|	M	M	m	M	B					1	C
N	⊤	N	N	n	N		A			2		C
O	○	O	O	o	O	B		8				C
P	*	P	P	p	P	B	A		4			
Q	?	Q	Q	q	Q	B	A		4	2		C
R	ρ	R	R	r	R	B			4		1	
S	⌈	S	S	s	S	B		8			1	
T	∼	T	T	t	T		A					
U	↓	U	U	u	U		A			2	1	
V	∪	V	V	v	V	B				2	1	
W	ω	W	W	w	W	B		8		2	1	C
X	⊃	X	X	x	X		A				1	C
Y	↑	Y	Y	y	Y	B	A	8			1	C
Z	⊂	Z	Z	z	Z			8		2		C
'	:	'	'	'	'	B				2		C
[	(	!	]	!	]	B						
,	;	,	..	,	..	B	A		4	2	1	
]	)	.	.	.	.	B			4			C
←	→	;	:	;	:	B	A		4		1	C
×	÷	=	+	=	+	B	A			2		
+	—	—	—	—	—	B	A	8		2	1	
/	\	/	?	/	?	B	A	8				
1	¨	1	+	1	+						1	
2	¯	2	@	2	@					2		
3	<	3	#	3	#					2	1	C
4	≤	4	$	4	$			8				
5	=	5	%	5	%				4			
6	≥	6	¢	6	¢				4	2		C
7	>	7	&	7	&				4		1	C
8	≠	8	*	8	*				4	2	1	
9	∨	9	(	9	(			8		2	1	
0	∧	0	)	0	)			8			1	C

↑
These characters appear in the communications diagnostic buffer.

[1] The character 9 can also indicate a ⒟ in the diagnostic buffer.

SERIAL I/O CHARACTER CODE CHARTS

The serial I/O character code charts show the characters that
are used with the serial I/O feature for 5-, 6-, and 7-bit codes.
Use these charts to determine the equivalent 5100 characters,
the bit values of the characters, and the column and row
designations of the characters.

Note: These charts show the default value of the characters
in the translation tables. The user can change the graphics
in the table; however, the bit value for a corresponding
column and row designation will not change. See the
IBM 5100 Serial I/O Adapter Feature User's Manual for an
explanation of how to change the characters in the trans-
lation tables.

Serial I/O 5-Bit Character 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	∩	∩		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	..

Serial I/O 6-Bit Character Chart

COL/ROW LOWER SHIFT	COL/ROW UPPER SHIFT	APL LOWER SHIFT	APL UPPER SHIFT	BASIC LOWER SHIFT	BASIC UPPER SHIFT	B	A	8	4	2	1	C	EBCD LOWER SHIFT	EBCD UPPER SHIFT
000	400	SPACE	SPACE	SPACE	SPACE							C	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	⎕	)	#	¨			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	←	→	@	ç		A						@	ç
101	501	/	\	/	?		A				1	C	/	?
102	502	S	⌈	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	∪	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	⊂	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	J̲	B 1 C	j	J̄
202	602	K	'	K	K̲	B 2 C	k	K
203	603	L	▯	L	L̲	B 2 1	l	L
204	604	M	|	M	M̲	B 4 C	m	M
205	605	N	⊤	N	N̲	B 4 1	n	N
206	606	O	○	O	O̲	B 4 2	o	O
207	607	P	∗	P	P̲	B 4 2 1 C	p	P
208	608	Q	?	Q	Q̲	B 8 C	q	Q
209	609	R	⍴	R	R̲	B 8 1	r	R
210	610	⍉	@	⊥	⊤	B 8 2		
211	611	[	(	⍕	!	B 8 2 1 C	$	!
212	612	⍉	⍉	⍉	⍉	B 8 4	RES	RES
213	613	⌿	⍀	⌿	⌿	B 8 4 1 C	NL	NL
214	614	↑	↓	↑	↑	B 8 4 2 C	BS	BS
215	615	\	\	\	\	B 8 4 2 1	IL	IL
300	700	×	÷	&	+	B A C	&	+
301	701	A	α	A	A̲	B A 1	a	A
302	702	B	⊥	B	B̲	B A 2	b	B
303	703	C	∩	C	C̲	B A 2 1 C	c	C
304	704	D	⌊	D	D̲	B A 4	d	D
305	705	E	∊	E	E̲	B A 4 1 C	e	E
306	706	F	_	F	F̲	B A 4 2 C	f	F
307	707	G	∇	G	G̲	B A 4 2 1	g	G
308	708	H	Δ	H	H̲	B A 8	h	H
309	709	I	∖	I	I̲	B A 8 1 C	i	I
310	710	&	⍙	Δ	∇	B A 8 2 C		
311	711	.	:	.	¬	B A 8 2 1	.	¬
312	712	⍢	⍢	⍢	⍢	B A 8 4 C	PF	PF
313	713	⊖	⊖	⊖	⊖	B A 8 4 1	HT	HT
314	714	⍦	⍦	⍦	⍦	B A 8 4 2	LC	LC
315	715	⍀	⍀	⍀	⍀	B A 8 4 2 1 C	DEL	DEL

Serial I/O 7-Bit Character Chart

Note: The 7-bit characters are shown in even parity.

COL/ROW	5100 CHARACTER	P	1	2	3	4	5	6	7	ASCII CHARACTER	COL/ROW	5100 CHARACTER	P	1	2	3	4	5	6	7	ASCII CHARACTER
000	α	0	0	0	0	0	0	0	0	NUL	200		1	0	1	0	0	0	0	0	SP
001	¥	1	0	0	0	0	0	0	1	SOH	201	!	0	0	1	0	0	0	0	1	!
002	τ	1	0	0	0	0	0	1	0	STX	202	..	0	0	1	0	0	0	1	0	"
003	⊥	0	0	0	0	0	0	1	1	ETX	203	#	1	0	1	0	0	0	1	1	#
004	⊥	1	0	0	0	0	1	0	0	EOT	204	$	0	0	1	0	0	1	0	0	$
005	φ	0	0	0	0	0	1	0	1	ENQ	205	%	1	0	1	0	0	1	0	1	%
006	θ	0	0	0	0	0	1	1	0	ACK	206	&	1	0	1	0	0	1	1	0	&
007	⊕	1	0	0	0	0	1	1	1	BEL	207	'	0	0	1	0	0	1	1	1	'
008	⊠	1	0	0	0	1	0	0	0	BS	208	(	0	0	1	0	1	0	0	0	(
009	⊃	0	0	0	0	1	0	0	1	HT	209	)	1	0	1	0	1	0	0	1	)
010	∩	0	0	0	0	1	0	1	0	LF	210	*	1	0	1	0	1	0	1	0	*
011	∪	1	0	0	0	1	0	1	1	VT	211	+	0	0	1	0	1	0	1	1	+
012	∨	0	0	0	0	1	1	0	0	FF	212	,	1	0	1	0	1	1	0	0	,
013	⊂	1	0	0	0	1	1	0	1	CR	213	--	0	0	1	0	1	1	0	1	-
014	♠	1	0	0	0	1	1	1	0	SO	214	.	0	0	1	0	1	1	1	0	.
015	ψ	0	0	0	0	1	1	1	1	SI	215	/	1	0	1	0	1	1	1	1	/
100	∇	1	0	0	1	0	0	0	0	DLE	300	0	0	0	1	1	0	0	0	0	0
101	↑	0	0	0	1	0	0	0	1	DC1	301	1	1	0	1	1	0	0	0	1	1
102	→	0	0	0	1	0	0	1	0	DC2	302	2	1	0	1	1	0	0	1	0	2
103	↓	1	0	0	1	0	0	1	1	DC3	303	3	0	0	1	1	0	0	1	1	3
104	←	0	0	0	1	0	1	0	0	DC4	304	4	1	0	1	1	0	1	0	0	4
105	×	1	0	0	1	0	1	0	1	NAK	305	5	0	0	1	1	0	1	0	1	5
106	÷	1	0	0	1	0	1	1	0	SYN	306	6	0	0	1	1	0	1	1	0	6
107	¨	0	0	0	1	0	1	1	1	ETB	307	7	1	0	1	1	0	1	1	1	7
108	|	0	0	0	1	1	0	0	0	CAN	308	8	1	0	1	1	1	0	0	0	8
109	A	1	0	0	1	1	0	0	1	EM	309	9	0	0	1	1	1	0	0	1	9
110	°	1	0	0	1	1	0	1	0	SUB	310	:	0	0	1	1	1	0	1	0	:
111	o	0	0	0	1	1	0	1	1	ESC	311	;	1	0	1	1	1	0	1	1	;
112	[]	1	0	0	1	1	1	0	0	FS	312	<	0	0	1	1	1	1	0	0	<
113	∈	0	0	0	1	1	1	0	1	GS	313	=	1	0	1	1	1	1	0	1	=
114	\	0	0	0	1	1	1	1	0	RS	314	>	1	0	1	1	1	1	1	0	>
115	ρ	1	0	0	1	1	1	1	1	US	315	?	0	0	1	1	1	1	1	1	?

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

READER'S COMMENT FORM

IBM 5100   SY31-0429-2
Communications/Serial I/O
Maintenance Information Manual

YOUR COMMENTS, PLEASE . . .

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

Page   *Comment*

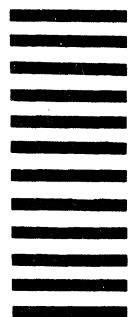I would like a reply. ☐

Name _____

Address _____

● Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

SY31-0429-2

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
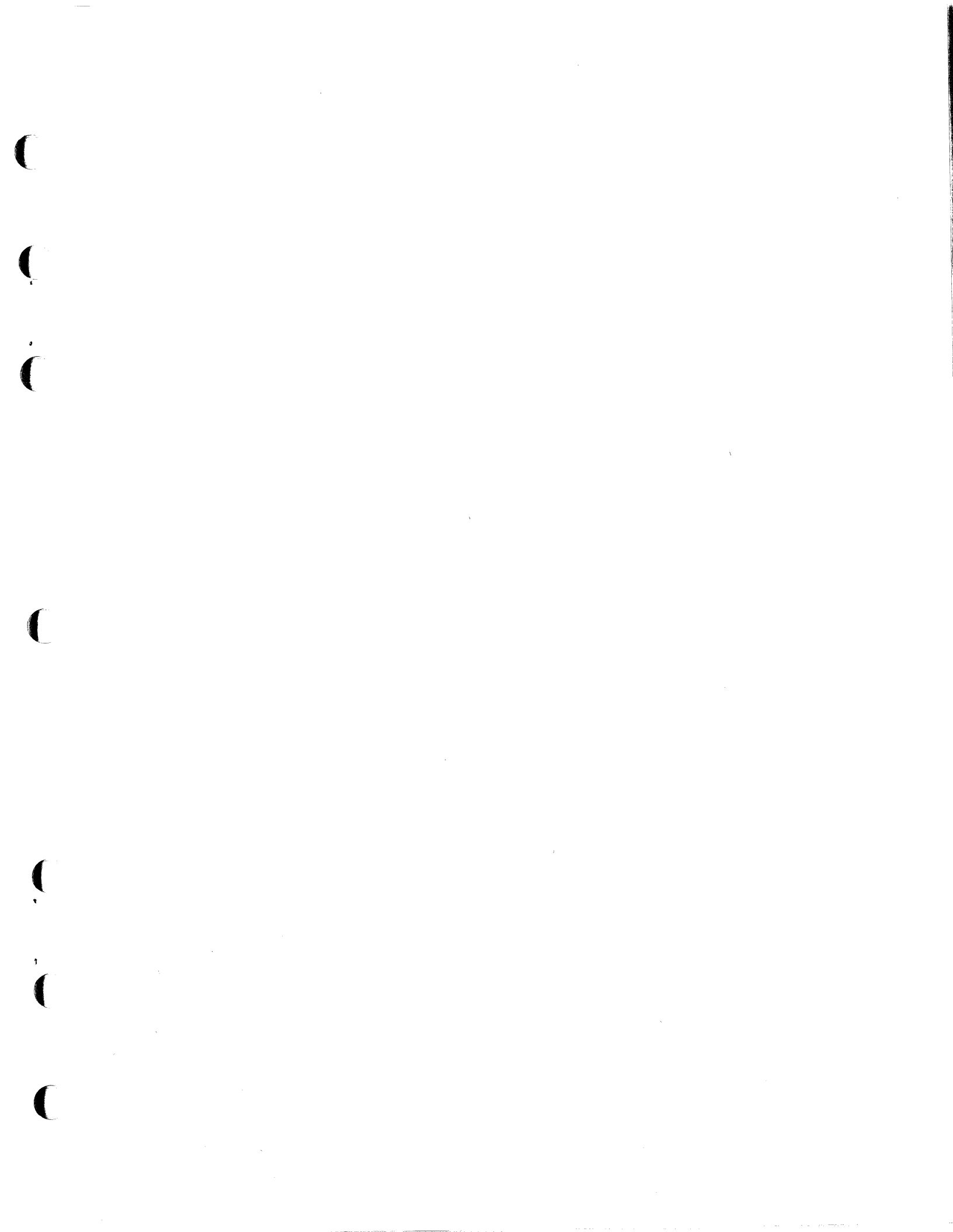Publications, Dept. 245
Rochester, Minnesota 55901

Fold   Fold

IBM

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)

IBM