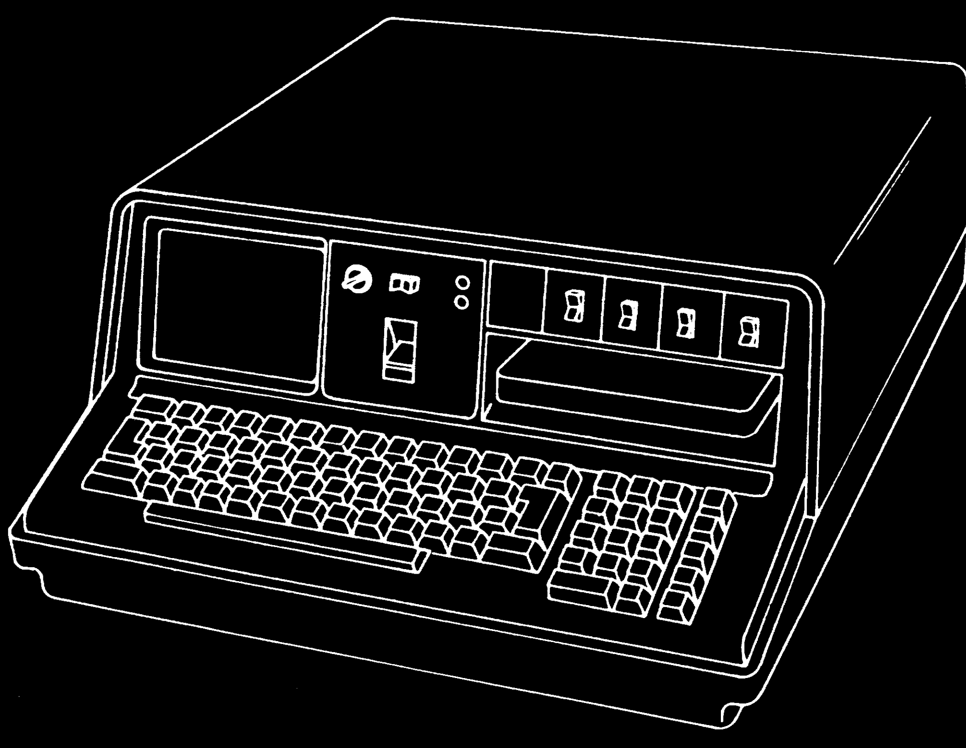
IBM

IBM 5110
Asynchronous Communications Feature
Maintenance Information Manual

5110

IBM 5110
Asynchronous Communications Feature
Maintenance Information Manual



SY31-0557-0

Preface

This manual is for personnel who service the IBM 5110 Asynchronous Communications feature. To use this manual, you should know how to operate the 5110 with the Asynchronous Communications feature, and you should understand basic start-stop data communications concepts.

Note: This manual follows the convention that *he* means *he or she.*

Related Publications

- *IBM 5110 Asynchronous Communications Feature User's Manual,* SA21-9314
- *IBM 5110 Computer Maintenance Information Manual,* SY21-0550
- *IBM 5110 System Logic Manual,* SY31-0552
- *IBM 5110 Asynchronous Communications Feature Reference Card,* GX21-9315

First Edition (January 1978)

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

Requests for copies of IBM publications should be made to your IBM representative or the IBM branch office serving your locality.

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

Contents

Manual Organization

The *Introduction* is an overview of the Asynchronous Communications feature for the IBM 5110. This section describes briefly the procedure that the user follows before the feature can be used.

The *Theory of Operation* section describes the function of the major components of the feature and the data flow between these components while the feature is in use.

The *Diagnostic Aids* section describes the aids that are available for diagnosing problems with the Asynchronous Communications feature. This section includes instructions for using these diagnostic aids.

The *Communications Networks* section reviews communications networks as related to the 5110 Asynchronous Communications feature. Included are the definitions of some commonly used terms and a brief description of modem operations as they apply to the Asynchronous Communications feature.

The *Appendixes* contain code translation charts and sequence charts that apply directly to the Asynchronous Communications feature.

The Asynchronous Communications feature provides the 5110 with start-stop communications capability through an EIA RS232C/CCITT V.24 - V.28[1] standard interface to a modem (data set). To a remote system, the 5110 appears to be the same as an IBM 2741 Communications Terminal. However, the 5110 can print, display, and write to tape or diskette both transmitted and received data, and it can transmit data from tape, diskette, and the keyboard.

The Asynchronous Communications feature consists of a logic card, an internal cable and backpanel connector, an external cable, and a program tape or diskette. The logic card is shared with the Serial I/O Adapter feature and is called the *asynchronous comm/serial I/O* card. The internal cable and the backpanel connector form a unit that couples the control lines, the signal lines, and the signal ground line between the logic card and the external cable. The external cable connects the 5110 to a modem (data set).

The Asynchronous Communications feature program tape or diskette contains a set of seven microprograms. A selected microprogram must be loaded into read/write storage before the feature can be used.

The user loads the microprogram by using the link command: LINK for BASIC and)LINK for APL. (The user specifies file 1, named MODECOM, in the Asynchronous Communications program tape or diskette.) The system loads a loader program and then prompts the user to select the line code and I/O device from a option table. The options follow:

Option Number	**Line Code**	**I/O Device**
1	EBCD	Tape
2	EBCD	Printer
3	EBCD	Diskette
4	Correspondence code	Tape
5	Correspondence code	Printer
6	Correspondence code	Diskette

[1]Electronics Industry Association recommended standard 232C and Consultive Committee for International Telephone and Telegraph recommendations V.24 and V.28.

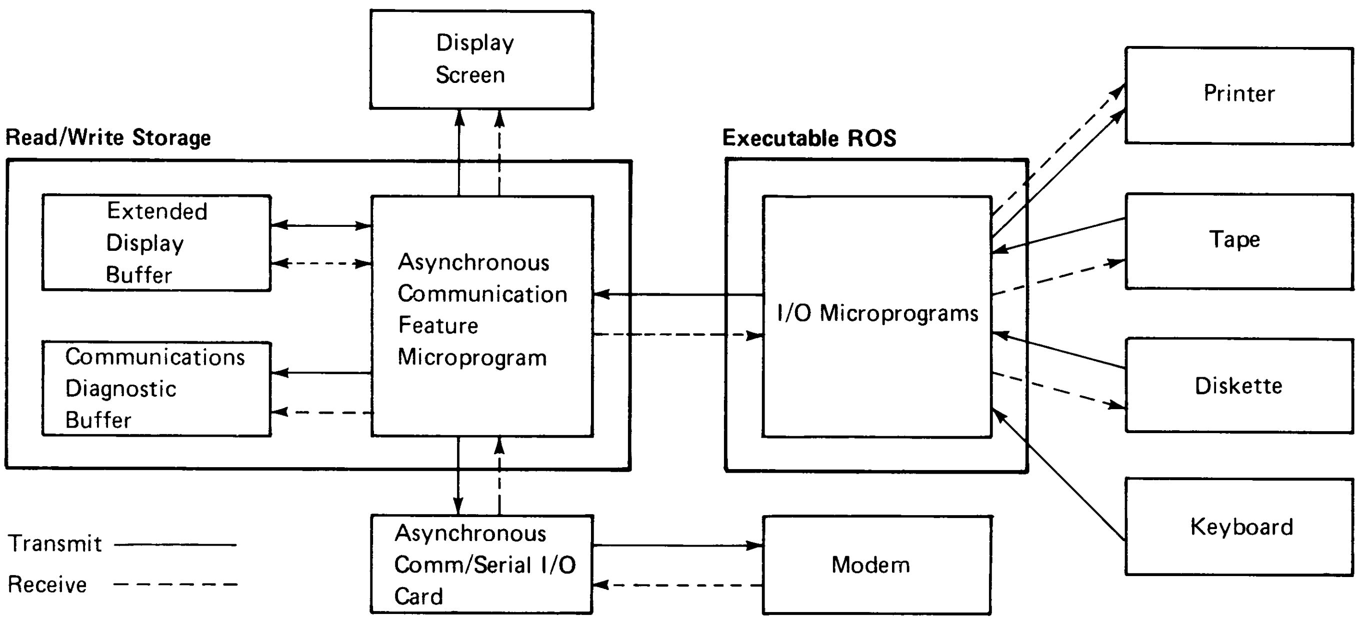
Although options 2 and 5 specify the printer, the printer can be used with all options, 1 through 6, as can the keyboard and the display. Support for tape operations and diskette operations is provided only in the microprograms specified (options 1 and 4 for tape and options 3 and 6 for diskette).

After a microprogram from the Asynchronous Communications feature tape or diskette is loaded into read/write storage, the microprogram controls the system. Normal APL and BASIC language operations can no longer be performed.

The user must next use communications commands (&SYSTEM and &RATE) to configure his system for graphics translation and to match the remote system for transmission rate. The user can then prepare a record offline for transmission, or can call the system at the remote location to send and receive data. See the *IBM 5110 Asynchronous Communications Feature User's Manual* for explicit information about how to load the microprogram and how to use the Asynchronous Communications feature.

MICROPROGRAM FUNCTIONS AND DATA FLOW

The following diagram shows the location of the microprogram and the data flow for asynchronous communications transmit and receive operations. Refer to this diagram while reading the description of the microprograms and data flow.



During asynchronous communications, the system functions as a communications terminal, and normal APL or BASIC language operations cannot be performed.

During asynchronous communications, the system uses a microprogram in read/write storage and microprograms in executable ROS.

The microprograms in executable ROS that are used during communications are the I/O microprograms that normally control the keyboard, the printer, tape operations, and diskette operations.

The microprogram in read/write storage is loaded from the Asynchronous Communications feature tape or diskette. While the system (functioning as a terminal) is offline, the asynchronous communications microprogram performs the following functions:

- Responds to operator control from the keyboard.
- Controls the format of information that is displayed, printed, and/or written to tape or diskette.
- Interacts with the I/O microprograms in executable ROS to exchange data and control information between the processor and I/O devices.
- Updates the extended display buffer in read/write storage. (See *Extended Display Buffer* in this section.)

When the 5110 is functioning as a terminal online, the asynchronous communications microprogram performs the following additional functions. The microprogram:

- Translates transmitted characters from internal code (EBCDIC) to line code and received characters from line code to EBCDIC.
- Inserts line control and function characters for transmit, and responds to line control and function characters during receive operations.
- Updates the communications diagnostic buffer in read/write storage. (See *Communications Diagnostic Buffer* in this section. The communications diagnostic buffer is not available with diskette options.)
- Calculates parity and inserts the parity bit (if required) for transmitted characters, and checks parity of received characters.
- Frames transmitted characters with the synchronizing bits (start and stop), and strips the synchronizing bits from received characters.
- Serializes transmitted characters and passes the character bit by bit to the asynchronous comm/serial I/O card, and deserializes bits received from the asynchronous comm/serial I/O card into characters.

Extended Display Buffer

The extended display buffer is built and maintained in read/write storage by the asynchronous communications microprogram. You can think of the extended display buffer as a shift register into which each character is entered at the right after all previously entered characters have been shifted to the left one character position. The character sequence resulting in the extended display buffer is the last entered character at the right and the first of the series toward the left.

Communications commands and all data that is transmitted and received are assembled in the extended display buffer by the microprogram. Other than the commands, only data characters are stored in the extended display buffer with the exception of new-line function characters, which are used to separate successive lines of data within a record. All the data contained in the extended display buffer can be displayed (on the CRT). The microprogram formats the data so that each line on the display represents a line of data. (Each line of data ends with a new-line character, which is not displayed.) A line longer than 64 characters (the width of the display screen) is effectively extended to the right of the display screen. By using the keyboard, you can scroll the displayed data to display any portion of the data in the extended display. While displayed, a line of data can be edited (changed or deleted). Changing or deleting any data within a line deletes all data that follows the edited line in the extended display buffer.

The contents of the extended display buffer can be printed, written to tape or diskette (if the tape option or diskette option was loaded), and transmitted.

The size of the extended display buffer in a 5110 with 16K bytes of read/write storage is 1K characters. A system with 64K bytes of read/write storage has an extended display buffer of 49K characters

Communications Diagnostic Buffer

The communications diagnostic buffer is a 256-byte buffer that is maintained by the asynchronous communications microprogram. Like the extended display, the communications diagnostic buffer is in read/write storage and resembles a shift register of character positions.

Every character that is transmitted and received, including line control and function characters, is stored in order of occurrence. Thus, the contents of the communications diagnostic buffer provides a history of the last 256 characters that were transmitted and/or received.

Characters are stored in the communications diagnostic buffer in line code with a flag bit stored with each character to signify direction (transmitted or received). When the contents of this buffer are displayed (see *Communications Diagnostic Buffer Display*), the asynchronous communications microprogram translates the characters and formats the display. The communications diagnostic buffer is not available with the diskette options (options 3 and 6).

Transmit

During transmit operations, data can be supplied from the keyboard, the extended display, tape, or diskette.

The following sequence describes a typical transmission of a character that originates at the keyboard. The sequence can be thought of as having two parts. The first part builds the extended display buffer, and the second transmits from the extended display buffer.

Building the Extended Display Buffer

Data is assembled in the extended display buffer as follows:

1. A data character key on the keyboard is pressed.

2. The I/O microprograms in executable ROS pass the character to the asynchronous communications microprogram in read/write storage.

3. The asynchronous communications microprogram updates the extended display buffer and places the character on the display screen.

Transmitting from the Extended Display Buffer

The asynchronous communications microprogram does the following:

1. Reads the character from the extended display buffer and translates the character from internal code (EBCDIC) to line code (either EBCD or correspondence code as specified by the option selected)

2. Updates the communications diagnostic buffer with the character in line code (options 1, 2, 4, and 5 only)

3. Calculates the parity of the character and adds the parity bit if required

4. Frames the character with the synchronizing (start and stop) bits

5. Serializes the character and passes it, start bit first, to the asynchronous comm/serial I/O card for transmission to the modem

Transmitting from Tape or Diskette

If the data for transmission comes from tape or diskette, the following sequence applies:

1. The user initiates the action with an &TAPEIN or &DISKIN communications command at the keyboard.

2. The I/O microprograms pass the command information to the asynchronous communications microprogram in read/write storage.

3. The asynchronous communications microprogram generates control information for a tape or diskette read operation and then forwards the control of the processor to the I/O microprograms in executable ROS.

4. The I/O microprograms read a physical record[1] from the tape or diskette and store the data that is read in a data buffer in read/write storage, then pass control back to the asynchronous communications microprogram.

5. The asynchronous communications microprogram then moves an entire line of data (one character at a time) from the data buffer to the extended display buffer.

6. The asynchronous communications microprogram starts with the first character of the line that was just assembled in the extended display buffer, and transmits each character from the extended display buffer as described above (see *Transmitting from the Extended Display Buffer*). Following the transmission of each line, the microprogram changes the 5110 to receive mode. After the received message is processed, transmission resumes. Alternation between transmit mode and receive mode continues until the entire file is sent. (Steps 5 and 6 or steps 4, 5, and 6 are repeated to transmit the entire file.)

---

[1]The length of a physical record for tape is 512 bytes. For diskette, a physical record can be 128 bytes, 256 bytes, or 512 bytes.

Data can be assembled in the extended display buffer so that a full record can be prepared before the first data character is transmitted. The data can be entered character by character while the system is in the EDIT state. (See *Composing Messages Offline* in the *IBM 5110 Asynchronous Communications Feature User's Manual.*) An attempt to transmit more than one line from the extended display buffer results in the deletion of all lines following the line transmitted. Normally, the user writes the record (a number of lines) to tape or diskette, and then sends from tape or diskette. However, you can send all lines directly from the extended display by sending the last line entered first. Then send the next-to-last line second, which deletes the line already sent, and so forth.

Receive

All data received by the system is stored in the extended display buffer and displayed on the screen, and can be printed and/or written to tape or diskette. Again, the operation can be split into two major functions: building the extended display buffer with the received data, and distributing the data from the extended display buffer to the I/O devices.

The following sequence describes a typical receive operation for a data character:

1. Bits from the modem (data set) are received by the asynchronous comm/serial I/O card.
2. The asynchronous communications microprogram collects the bits from the asynchronous comm/serial I/O card and deserializes the bits into characters.
3. The asynchronous communications microprogram strips the synchronizing bits (start and stop) from the character.
4. The asynchronous communications microprogram checks the parity of the character.
5. The asynchronous communications microprogram updates the communications diagnostic buffer (options 1, 2, 4, and 5 only).
6. The asynchronous communications microprogram translates the character from line code (EBCD or correspondence code) into EBCDIC (internal code) and stores the data character in the extended display buffer.
7. The asynchronous communications microprogram formats and displays the data character.

With options 2 or 5, if the data received is printed, the asynchronous communications microprogram assembles a complete line of data in the extended display buffer and transfers control to the I/O microprograms to print the data. With options 1, 3, 4, and 6, if the data is printed and/or written to tape or diskette, the asynchronous communications microprogram assembles a complete transmission in the extended display buffer and receives an end-of-transmission character before it moves data into a data buffer in read/write storage. When the buffer is full, the asynchronous communications microprogram transfers the control information to the I/O microprograms in executable ROS, which write the data to tape or diskette and/or the printer. When the length of a transmission exceeds the size of the extended display buffer, an error code is displayed, and operator intervention is required. See the *IBM 5110 Asynchronous Communications Feature User's Manual* for information about error codes and user responses.

DATA FORMAT

The system uses the start-stop format for the Asynchronous Communications feature. In the start-stop format, each character is synchronized separately on the data line. To synchronize each character, a start bit (ST) is added to the beginning of the character and a stop bit (SP) is added to the end of the character. The start bit always changes the transmit data line level from the quiet state level (mark) to a space level. Data bits are transmitted and received at mark level. The stop bit returns the line to a mark level at the end of a character. The line remains at the mark level until the next start bit.

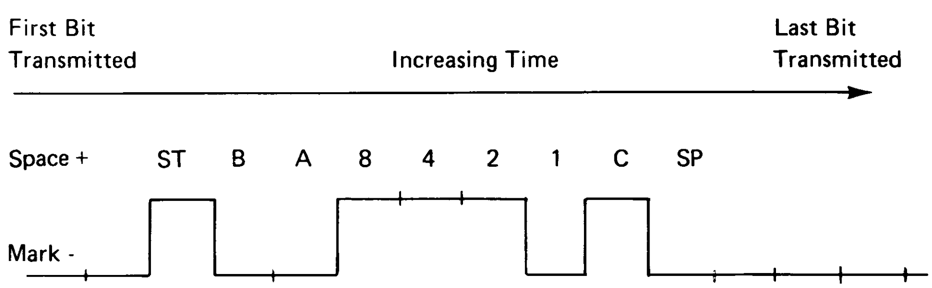
The code used with the Asynchronous Communications feature is either EBCD (extended binary coded decimal) or the correspondence code.

EBCD and correspondence codes each contain 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 different combinations of bits indicate different characters. Bit C is a check bit (parity bit) and is used to maintain odd parity of each character for error-checking purposes.

Additional character combinations are obtained with a shift function character preceding a character or characters. For example, if the character codes are preceded by an up-shift function character, the character codes represent upper-shift characters. Likewise, if the character codes are preceded by a down-shift function character, the codes represent lower-shift characters. The character represented by a particular code is dependent upon the translation selected by the user with the &SYSTEM command and the line code option used. For more information on shift function characters, see *Shift Function Characters* and *Communications Diagnostic Buffer Display*.

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 (-3.0V to -25.0V).

The absence of a data bit is indicated by a space; a space is a plus (+) voltage level (+3.0V to +25.0V). The following graph shows how the character A would look on the data line with the start (ST) and stop (SP) bits added:



EBCD Character A (bits B, A, and 1) Odd Parity[1]

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

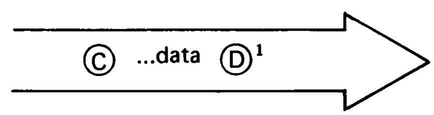
COMMUNICATIONS CONTROL CHARACTERS

Messages are enclosed between 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 5110, and by the remote system as shown in the following diagram (the arrow indicates the direction of data flow):

Action	5110	Remote System

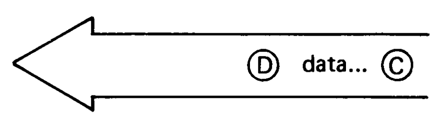
The lines 'clear to send' and 'data set ready' are activated by the modem. (This is indicated on the 5110 display by the up-arrow (↑) on the status line.)

The 5110 waits approximately 1 second after 'clear to send' and 'data set ready' become active, then starts sending the first message to the remote system.

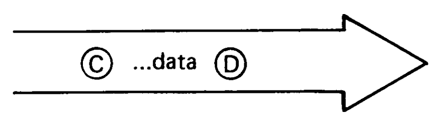


The 5110 completes the message and waits for the remote system to transmit.

The remote system responds.



After a 66ms delay (at 134.5 bps or approximately 50ms at 300 bps), the 5110 changes to transmit state and sends the next message. (These delays are needed to simulate the timings of the IBM 2741.)



---

[1]For switched communications facilities (dial up), the Ⓓ is transmitted by the asynchronous communication microprogram immediately after the approximately 1-second delay, which is indicated by the arrow in the status line changing from up (↑) to right (→). The 1-second delay allows the communications lines to stablize. For leased lines, the transmission of the Ⓓ immediately precedes the transmission of the first data character or function character.

When the 5110 is in XMIT state, pressing the EXECUTE key transmits a new-line function character followed by the end-of-transmission control character (Ⓒ). Pressing the ATTN key transmits only the Ⓒ.

If the 5110 is in the receive state, and an end-of-transmission control character (Ⓒ) from the remote system is lost (because of line noise or some other reason), the 5110:

- Remains in the receive state until a Ⓒ is received
- Interprets a received Ⓓ control character as a data character and continues to store any received data characters in the extended display buffer
- Goes to the transmit state when a Ⓒ is received
- Leaves the receive state when the CMD and ATTN keys are pressed (which causes the 5110 to go to the transmit state and send a Ⓓ)

If the 5110 is in the receive state, and a start-of-transmission control character (Ⓓ) from the remote system is lost, the 5110:

- Ignores all data characters and function characters until a character with the same code as the Ⓓ is received. (The bit configuration of the Ⓓ control character is not unique and is used to represent data characters in both EBCD and correspondence code.)
- Goes to transmit state when a Ⓒ is received from the remote system.

COMMUNICATIONS FUNCTION CHARACTERS

Function characters are special characters that the 5110 and the remote system use during communications to control specific functions. These characters, their functions, and their bit values are shown under *Code Chart of Function and Control Characters* in Appendix A.

Following is a more detailed explanation of the function characters that apply to the 5110.

Shift Function Characters

Two methods are used by the 5110 to produce shift function characters (up-shift and down-shift). The first and usual method is under the complete control of the asynchronous communications microprogram and occurs during the translation of the EBCDIC character from the extended display buffer into line code. The microprogram keeps track of the upper-shift/lower-shift status, which is set by the last shift function character transmitted. (Default is lower shift.) When the character code from the extended display is found to be in the opposite shift from the shift status, the microprogram precedes the transmission of the data character with the appropriate shift function character and updates the shift status. For example, when the microprogram reads a character from the extended display buffer and finds the translation for the character is upper shift in the translate table, the microprogram checks the shift status. If the shift status is lower shift, the microprogram forces an up-shift function character to be transmitted prior to the transmission of the translated data character. If the shift status is upper shift, no shift function character is sent before the line code representing an upper-shift data character is sent. The same logic is used for lower-shift characters.

The second method of transmitting shift function characters requires operator intervention and is used primarily during sign-on procedures with a remote system. When the shift function key is pressed (CMD and alphameric 4 keys), the 5110 transmits a shift function character. If the shift status when the keys are pressed is lower shift, the up-shift function character is transmitted and the shift status is changed. If the keys are pressed again, the down-shift function character is sent. This sequence produces the up-shift, down-shift character sequence that is required by some sign-on procedures. This key combination (CMD and alphameric 4) does not lock the keyboard in any shift, and it does not replace the shift keys on the keyboard for producing upper-shift characters in the translation specified by the &SYSTEM command.

During receive operations, the 5110 uses the received shift function characters to update shift status. The shift status is used to select the appropriate portion of the translate table to use for received data characters when received characters are stored in the extended display.

The shift function characters, up-shift (indicated by a j in the diagnostic buffer display) and down-shift (indicated by a ⎕ in the diagnostic buffer) indicate to the remote system and the 5110 that the data following is upper or lower shift (see *Communications Diagnostic Buffer Display*). For example, if the 5110 is receiving and the communications diagnostic buffer contains

jABCDEF⎕GHIJK

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

Backspace Function Character

The backspace function character is transmitted when the backspace key is pressed while the display indicates HOME state or XMIT state. A backspace function character (indicated by a ≠ in the diagnostic buffer) is an indication to the 5110 and the remote system to backspace 1 character. For example, if the 5110 received a character sequence that appeared as follows in the communications diagnostic buffer display:

A≠jF

the 5110 would display the character A, backspace the cursor, and display the character _ (an upper-shift F) to form the character A̲ (A underscore) in the extended display.

Tab Function Character

A tab function character (indicated by a ϵ in the diagnostic buffer) is an indication to the 5110 and remote system to tab to the next tab setting. For example, if the 5110 received

ABCDϵEFGH

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 ↑ and a ↓ respectively in the diagnostic buffer) set and restore the bypass function on the 5110 when the characters are received. When the bypass function is set (bypass function character received), the 5110 does not display, write, or print data.

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 5110 does not transmit the bypass and restore function characters.

New-Line and Line-Feed Function Characters

A new-line function character (indicated by an o in the communications diagnostic buffer) indicates the end of a line of data. For example, if the 5110 received

ABCDEoFGHIo

the display would show

ABCDE

FGHI

__ (cursor)

The new-line function character is transmitted and received by the 5110. The new-line character is transmitted followed by an end-of-transmission character when the EXECUTE key is pressed and the system is in XMIT state. The new-line function character is also supplied by the asynchronous communications microprogram and stored in the extended display when messages are assembled offline. When variable length records from the extended display buffer are written to tape or diskette for later transmission, the new-line character is also written. When fixed-length record-oriented files are transmitted from diskette, the microprogram provides a new-line character following each record.

A line-feed function character (indicated by a b in the diagnostic buffer) is only received by the 5110; the 5110 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 5110 received

ABCDbEFGHb

the display would show

ABCD

EFGH

__ (cursor)

New-line and line-feed function characters also indicate to the 5110 that it should start the printer operation. Only complete lines of data are printed. A complete line is data ending with a new-line or line-feed function character. For example, if the following message is received:

ⒹABCoDEFoGHIⒸ

New-Line Characters

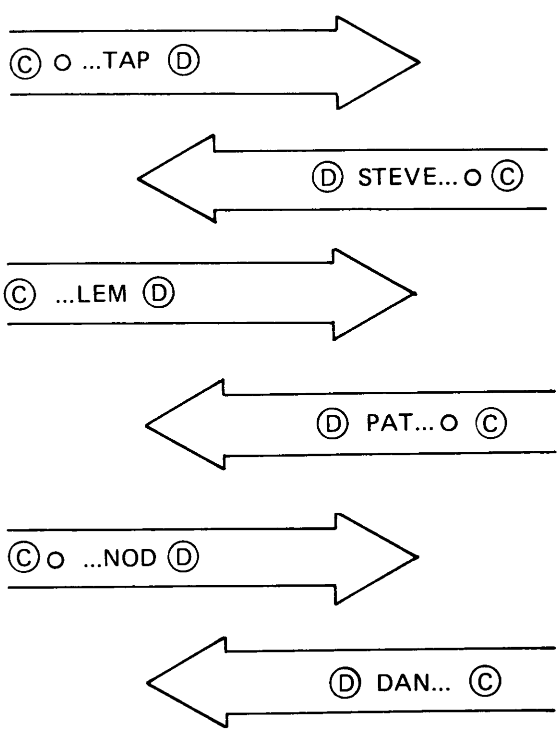
the last line of data (GHI) is not printed because it does not end with a new-line character; however, it is stored in the extended display.

Tape and diskette output operations start after the end-of-transmission character is received. Again, only complete lines of data are written.

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 received 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 5110 **Remote System**



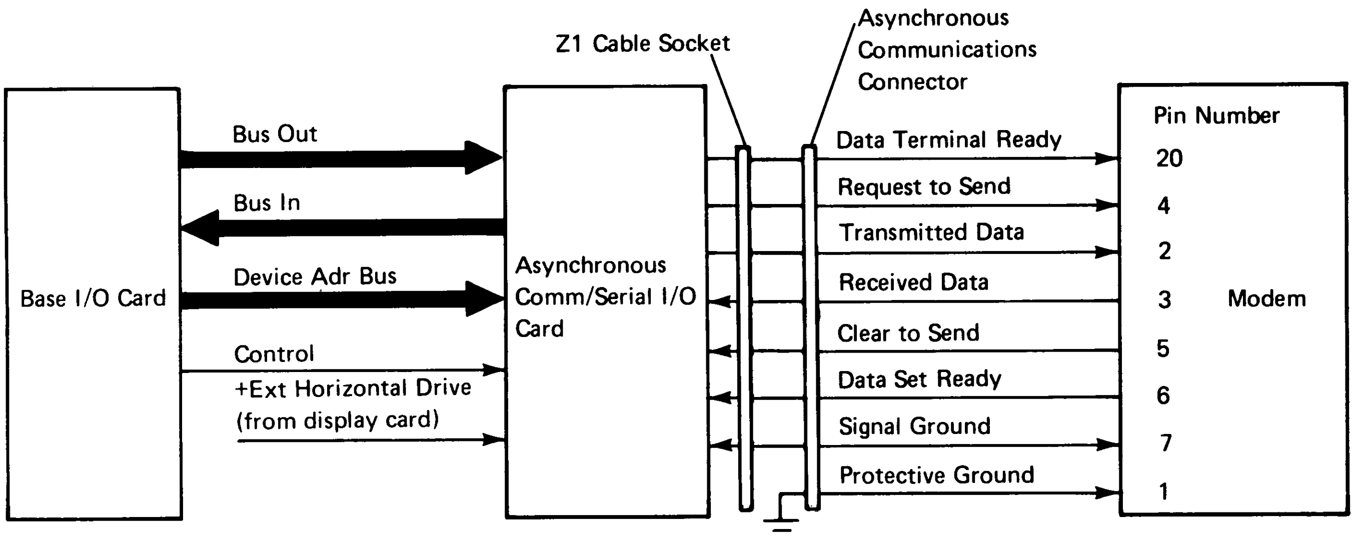
Printer at 5110	
No output	'PAT' is not printed because it is a transmitted message.
STEVE	'STEVE' is printed because it is a received message that ends with a new-line character.
No output	'MEL' is not printed because it is a transmitted message.
MELPAT	'MELPAT' is printed. 'PAT' is a received message that ends with a new-line character. Because 'MEL' was not ended with a new-line character, 'PAT' was added to the same line.
No output	'DON' is not printed because it is a transmitted message.
No output	'DAN' is not printed. Although 'DAN' is a received message, the new-line character is missing.

The message DAN is stored in the extended display buffer, but is not printed or written because a new-line or line-feed function character was not received.

If communications continue and the extended display buffer overflows before a complete line of data is received, the message DAN can be lost.

COMMUNICATIONS MODEM INTERFACE

The modem interface between the asynchronous comm/serial I/O card and the modem is the EIA RS232C/CCITT V.24-V.28 standard interface and is connected as shown in the following diagram:



The output EIA lines are turned on, and the input EIA lines are sensed, by the asynchronous communications microprogram. When an EIA control line is on, it is indicated by a plus voltage level (+3.0V to +25.0V). An off signal is indicated by a minus voltage level (-3.0V to -25.0V). Data lines (transmitted and received) are at a plus level (+3.0 to +25.0) for a space and a minus voltage level (-3.0 to -25.0) for a mark.

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

If the 5110 is going to transmit data:

1. The 5110 turns on 'data terminal ready' to indicate that it is ready to communicate.
2. The modem turns on 'data set ready' to indicate that it is ready to communicate.
3. The 5110 turns on 'request to send' to indicate that it is ready to send data to the modem.
4. The modem turns on 'clear to send' to indicate that it is ready to send data to the remote system.
5. The 5110 sends the data to the modem on the 'transmitted data' line. (See *Data Format* in *Theory of Operation*).

If the 5110 is going to receive data:

1. 'Data terminal ready' and 'data set ready' are on as in a transmit operation. ('Request to send' is also on. The system therefore requires a full duplex facility.)
2. The 5110 receives the data 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 5110. Refer to the *IBM 5110 System Logic Manual* (reference 425) for the communications connector and the asynchronous comm/serial I/O card pin numbers.

ASYNCHRONOUS COMM/SERIAL I/O CARD

The asynchronous comm/serial I/O card is the interface between the base I/O card and the modem. The card:

- Stores control information and the transmit data bit that is sent by the asynchronous communications microprogram
- Stores status information and the received data bit that is read by the asynchronous communications microprogram
- Provides level 1 interrupts to the asynchronous communications microprogram at a selectable rate for receive data sampling and transmit timing
- Contains drivers and receivers to convert the modem EIA voltage levels to system voltage levels
- Checks the device address and bus out for correct parity when the asynchronous communications/serial I/O card is addressed
- Monitors the 'comm received data' line during communications receive operations for a long space condition, and signals the processor via an interrupt when a long space occurs
- Selects the asynchronous communications external connector under the control of the microprogram

Refer to the asynchronous comm/serial I/O card logic diagram in the *IBM System Logic Manual* (reference 425) while reading the following description of the asynchronous comm/serial I/O card operation.

Addressing

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

DAF (device address F), '-control strobe', and '-bus out bit 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 asynchronous communications 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 bit 0'

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

'-bus out bit 2'

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

'-bus out bit 3'

The timer interrupt latch is reset.

'-bus out bit 4'

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

'-bus out bit 5'

The long space interrupt latch is reset.

'-bus out bit 6'

The 'comm data term ready' line to the modem and other controls within the card are set.

'-bus out bit 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 asynchronous communications feature microprogram to gate data and controls to the card as follows:

'-bus out bit 0'

A data bit is gated onto the '+comm transmit data' line.

'-bus out bit 5'

Rate data is gated to be loaded to the card from the asynchronous communications microprogram.

'-bus out bit 6'

Rate data is serialized to the asynchronous comm/serial I/O card (specifies the data rate clock preset value).

Start Execute

The '+start execute' line is made active by the asynchronous communications microprogram to gate data and status information that is stored on the card to the microprogram. Data and status is sent on the bus in lines as follows:

'bus in bit 1'

'Clear to send' is sensed. (Refer to *Communications Modem Interface*.)

'bus in bit 2'

'Data set ready' is sensed. (Refer to *Communications Modem Interface*.)

'bus in bit 4'

'Long space interrupt' defines the source of the program level 1 interrupt.

'bus in bit 7'

'Received data' is sensed. (Refer to *Communications Modem EIA Interface*.)

'bus in bit P'

Odd parity is maintained to the bus in circuitry from the asynchronous comm/serial I/O card.

External Horizontal Drive

The line '+ext horizontal drive' increments a counter that is used to detect a long space from the remote system. (See *Communications Transmit Interrupt* and *Receive Interrupt* in this section.)

Asynchronous Comm/Serial I/O Card Transmit Operation

When the asynchronous comm/serial I/O card and the modem are ready to transmit, the card is addressed, and the interrupt logic and data rate clock are enabled. When an interrupt occurs at a preselected rate (134.5 bps or 300 bps as specified by the user), the asynchronous communications microprogram sends the next bit to be transmitted to the card via '-bus out bit 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 '-interrupt 1 req' (interrupt request), which tells the asynchronous communications 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 bit 0'.

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

4. The bit is passed, via a driver circuit, to the '+comm transmit data' line going to the modem.

5. The timer interrupt latch is reset by the asynchronous communications 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 asynchronous communications microprogram disables the timer interrupt, but allows the clock to run, until the program is ready to send another character.

Asynchronous Comm/Serial I/O Card Receive Operation

When the 5110 is in the receive state, the '+comm received data' line is monitored by the asynchronous comm/serial I/O 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 5110 Computer Maintenance Information Manual* for a description of a program level 1 interrupt.) This interrupt tells the asynchronous communications microprogram that a bit has been received.

The microprogram addresses the asynchronous comm/serial I/O card and activates the '+start execute' line (which samples the bit on the '+received data' line instantaneously) to gate the bit to the asynchronous communications microprogram.

Line noise affects the value of a bit only if the noise occurs when the line is sampled for a bit. For example, if a noise pulse occurred at the same time as the '+start execute' 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 asynchronous communications microprogram enables the start bit error detect logic. This logic monitors the '+comm received data' line for a space signal equal to at least one-half the bit time of the data rate being used.

2. The modem receives the start bit from the communications line and sends the bit on the '+comm received data' line to the asynchronous comm/serial I/O card.

3. The '+comm received data' line is gated to 'bus in bit 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 asynchronous communications microprogram stops the clock and enables the start bit error detect logic ('+bus out bit 7') again.

4. The fall of the data rate clock pulse on '-ungated osc' sets the timer interrupt latch.

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

   Note: The sequence of steps 4 and 5 repeats until the microprogram determines that all bits of the character are received.

6. 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 5110 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 '+comm received data' line is monitored during transmitting for a continuous space greater than 63 ms. When a space level longer than 63 ms is detected, and '+com data set ready' and '+comm data term ready' are active, a program level 1 interrupt request is generated by the asynchronous comm/serial I/O card. The long space interrupt request sets the long space interrupt status on '+bus in bit 4' to the asynchronous communications microprogram. The asynchronous communications microprogram terminates the current transmission and changes from transmit to receive and waits for a communications control character (Ⓒ or Ⓓ). This response allows the remote system to gain control of the line. If a long space is detected while the the 5110 is receiving, an interrupt request is generated as in transmit, but it is ignored by the asynchronous communications microprogram.

Receive Interrupt

If the ATTN key on the 5110 is pressed during a communications receive operation, the asynchronous communications microprogram causes the asynchronous comm/serial I/O card to transmit a space level on the '+comm transmit data' line for about 200ms at 134.5 bps or for about 90ms at 300 bps. This space level is recognized as a transmit interrupt (long space interrupt) by the remote system. The 5110 remains in receive state until it receives an end-of-transmission character (Ⓒ).

ERROR DETECTION

The types of transmission errors detected by the 5110 during communications 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 asynchronous communications microprogram maintains the correct parity of each character transmitted, and checks for the correct parity of each character received. Both EBCD and the correspondence code use odd parity.

Parity errors are indicated by an OUT (O backspace U backspace T) character on the display screen during communications operations and by the character $\tilde{\wedge}$ in the communications diagnostic buffer.

Start Bit Errors

During receive operations, the received data line is monitored for a space bit (+voltage level). (When the 5110 or remote system is in transmit mode but is not actually transmitting data, control, or function characters, 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 whether 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 asynchronous communications microprogram checks each character for a stop bit by checking for the correct number of bits in each character. For example, an EBCD or a correspondence code character contains 9 bits, including the start and stop bits. Therefore, the ninth bit must be a stop bit or a stop bit error occurs.

Stop bit errors are indicated by an OUT character on the display screen and by $\tilde{\vee}$ in the communications diagnostic buffer.

Much of the hardware and the microprograms in executable ROS that are used during communications are also used in APL or BASIC language operations. You should run all the hardware diagnostic programs when troubleshooting a communications problem. These diagnostic programs are described in the *IBM 5110 Computer Maintenance Information Manual*.

A wrap connector (either P/N 1608894 or P/N 4360195) is required for the MDI (maintenance diagnostic integrated) program MDI 820. (In Japan, the wrap connector is replaced by a switch assembly that is part of the external cable.) The wrap connector can be attached to the end of the cable or to the 5110 communications connector. The wrap connector connects 'request to send' to 'clear to send, 'data terminal ready' to 'data set ready', and 'transmitted data' to 'received data'. This allows the communications diagnostic MDI 820 to check the control and data lines with the 5110 isolated from the modem. If the wrap connector is attached to the end of the modem cable, the cable can also be tested. Refer to the *IBM 5110 Computer Maintenance Information Manual* for the internal wiring of the wrap connectors.

If you suspect the internal cable, you can jumper the following lines together at the cable socket and then run MDI 820:

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

Refer to the *IBM 5110 System Logic Manual* (reference 425) for cable socket and pin locations.

COMMUNICATIONS DIAGNOSTIC BUFFER DISPLAY

Except for diskette options (options 3 and 6), all characters transmitted or received during communications are placed in the communications diagnostic buffer. The contents of the communications diagnostic buffer can be displayed and used to help isolate problems between the 5110 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 the 5110 is receiving as follows:

- Press and hold the CMD key and press the ATTN key. This procedure forces the HOME state. (The rest of a message is lost, and you may have to sign on or dial the remote system again to resume the receive operation.)

The receive operation stops, and you can display the last 256 characters that were transmitted and/or received. (See *Displaying the Communications Diagnostic Buffer*.)

You can stop the communications operation when you are transmitting data from the 5110 keyboard as follows:

1. Enter the data from the keyboard and press the EXECUTE key. The 5110 goes into receive state.
2. Press and hold the CMD key and press ATTN to force the HOME state.
3. You can now display the communications diagnostic buffer.

You can also stop the communications operation when the 5110 is transmitting data from tape as follows:

1. Press and hold the shift key and press the ATTN key. The 5110 stops transmitting at the end of a line of data and goes into receive state.
2. Press and hold the CMD key and press ATTN to force the HOME state.
3. You can now display the communications diagnostic buffer.

Displaying the Communications Diagnostic Buffer

To display the communications diagnostic buffer, enter the command &DIAG as follows:

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

2. Enter DIAG and press the EXECUTE key.

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

When the buffer is displayed:

- The characters in the buffer are displayed in four lines at the top of the display screen.

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

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

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

- Function and control characters are displayed as shown in *Code Chart of Function and Control Characters* in Appendix A. Except for the start-of-transmission characters (Ⓓ), the direction of all control and function characters is shown on the display by left arrows and right arrows beneath the character. A received function or control character is displayed above a left arrow (←). A transmitted function or control character is displayed above a right arrow (→). The start-of-transmission character appears as a data character and is underlined if it was transmitted, and not underlined if it was received. The start-of-transmission character can be easily differentiated from a data character by its position within the displayed message. Except for the first occurrence of the Ⓓ, it is immediately preceded by a Ⓒ. An example of function and control characters is shown on line 1.

- All data characters are displayed as lower-shift APL characters.

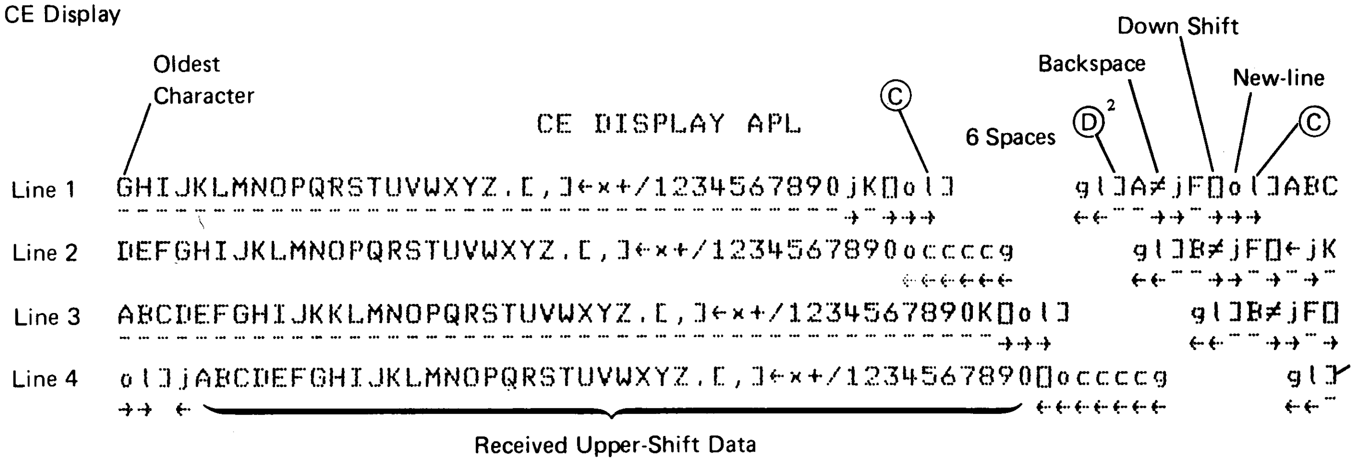
- Upper-shift data characters are also displayed as lower-shift characters preceded by an up-shift function character (j). An example of this is shown on lines 2 and 3.

Use the *Communications Code Charts* in Appendix A to decode the data characters in the communications diagnostic buffer display into the appropriate graphic characters. The display heading (CE DISPLAY APL or CE DISPLAY BASIC) indicates the translation to use.

To determine whether the displayed data is upper shift or lower shift, scan from right to left to the first shift function character. For example, in line 4 of the CE display, an up-shift function character (j) precedes the character A; therefore, the characters A through 0 are upper-shift characters. If a Ⓓ is encountered before a shift function character when you are scanning from right to left, the characters to the right of the Ⓓ are lower shift.

The following example of data in the communications diagnostic buffer display (*CE Display*) shows how data, function, and control characters appear when displayed. The example of data in *Extended Display* shows how the same data appears when the extended display buffer contents are formatted for the display screen.

CE Display



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

Character in Communications Diagnostic Buffer	Meaning
j	Up-shift
⌊	Ⓒ
o	New line
≠	Backspace
∊	Tab
⎕	Down-shift
](EBCD)9(correspondence code)	Ⓓ
Blank	Space

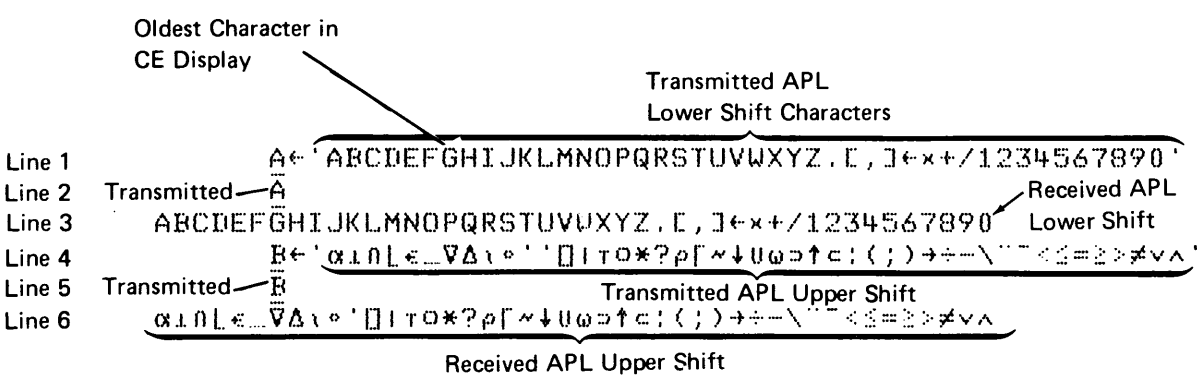
The characters in the chart at the left are the control and function characters that are both transmitted and received by the 5110.

Except for the Ⓓ control character, the direction (transmitted or received) of the characters is shown by arrows beneath the character.

← means received.
→ means transmitted.

The Ⓓ character appears as data (underscored if transmitted and not underscored if received).

Extended Display



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 the message (A←'ABCDEF) has overflowed the communications diagnostic buffer and is not displayed on the CE display. The end of this message is indicated by the first Ⓒ (I) on line 1.

Also on line 1 of the CE display is the message A̲ that was transmitted by the system 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 Ⓒ (I), which indicates the end of the message. The rest of the characters between the Ⓓ and the Ⓒ make up the message A̲.

The message A̲ was not transmitted as one character. This message is composed of the data and function characters A≠jF⎕ol 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 (≠) to be transmitted.

3. The shift key was pressed and held, and the character _ (F key) was entered. This caused the up-shift function character (j) and the data character _ to be transmitted. (The F character appears in the CE display because all data characters are displayed as lower-shift APL characters.)

4. The shift key was released and the EXECUTE key was pressed to indicate the end of the message. This caused the down-shift function character (⎕), the new-line character (o), 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 up-shift function character (j). 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 (how faithfully the network reproduces the signal input to the network at its output) through the analysis of a known signal pattern transmitted by a remote facility and received by the 5110. This provides the service personnel with a technique for isolating problems between the IBM 5110 and the communications network.

The network analysis subroutine samples and analyzes an alternating mark and space pattern of either 134.5 bps or 300 bps received by the 5110. The '+received data' line is sampled at a uniform rate of five samples per bit (Figure 3-1). The routine expects five samples of space level, followed by five samples of mark level, and so forth. Deviations from the expected results are called mark errors and space errors. The number of samples in error is accumulated in counters and is displayed at the end of the test. (See *How Data Is Accumulated*.) The total number of bits sampled (both marks and spaces) is also counted. In addition to mark samples and space samples, the network analysis subroutine checks consecutive bit relationships by comparing the position of a pulse from the asynchronous comm/serial I/O card to the five mark samples or space samples. Sampling of consecutive bits is synchronized by a hardware timer interrupt that ideally occurs at the midpoint of the bit (Figure 3-2). The hardware timer is synchronized every 20 bits on a space bit.

Synchronizing on the space bit uses start-bit-detect circuitry. With the start-bit-detect circuitry enabled, an interrupt occurs halfway through the space bit. If the signal is noisy and the interrupt does not occur within the allocated time period, a sync error is detected; one is added to an error counter, and the next space bit is tested as a sync bit. If 256 consecutive attempts to synchronize on space bits result in sync errors, 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 whether 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 are analyzed before the hardware timer is resynchronized. If one or more of the first five samples are bad, a sync error is accumulated; the center sample of the bit and 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 is not sampled.

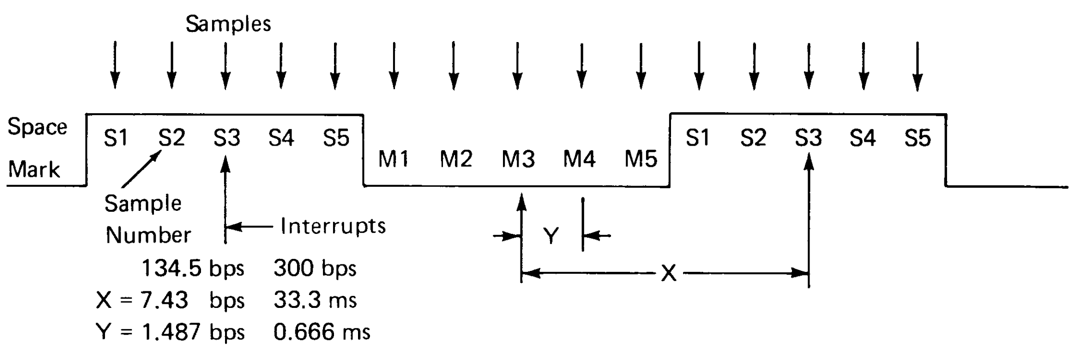


Figure 3-1. Sample Sequence

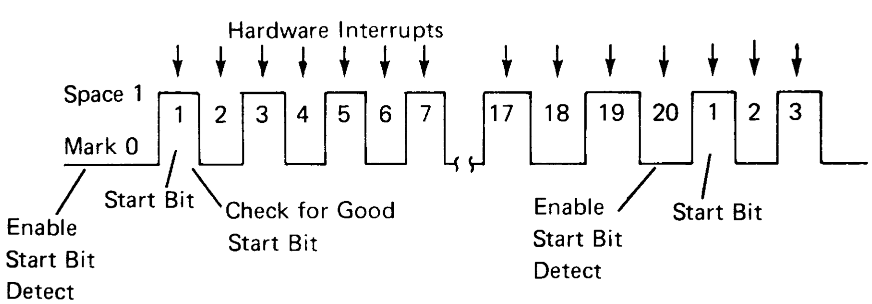


Figure 3-2. Synchronization Sequence

Remote End Requirements

The network analysis subroutine requires a continuous alternating 1/0 pattern on the 'received data' line. The bit rate may be either 134.5 bps or 300 bps.

The source of the alternating pattern may be:

- Remote system software routine
- CE test panel at the remote location
- Teleprocessing diagnostic analyzer tester (TDAT) at the remote location

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

An alternative procedure involves the use of the TDAT. With this procedure, the 5110 transmit routine provides an alternating pattern that is recorded on tape by the TDAT. The recorded data is then played back to the 5110 by the TDAT for the network analysis subroutine. Refer to the *1200 Teleprocessing Diagnostic Analyzer Tester* manual, S226-3029, for this procedure.

To transmit an alternating mark and space pattern on the 5110, set the system to loop on step 24 of MDI 895 for 300 bps. Loop on step 25 for 134.5 bps.

Operating Instructions

To operate the network analysis subroutine:

1. Load DSP (diagnostic supervisor program) and select MDI 890 from the DSP menu.
2. Select MDI 895 (Asynchronous Communications and Serial I/O Diagnostic Routines).
3. Enter 0 to display the MDI options. Then enter step 26 for 134.5 bps or step 27 for 300 bps. The display indicates the procedure for establishing a data link with the facility that will transmit the alternating test pattern.
4. Dial the remote facility and establish a voice link.
5. Coordinate the type of test to be performed (see *Remote End Requirements*) and the procedure to be used to establish the data link (channel).
6. Press the EXECUTE key to make the IBM 5110 ready and to initiate the subroutine.
7. Place the answering modem in the data mode by pressing the DATA key. (If you originated the call, the person on the other end of the line goes to data mode.)
8. When the carrier tone is heard in the originating data set, place it in data mode by pressing the DATA key on that modem.

The status displayed at the bottom of the display screen should proceed from DS NOT READY when the EXECUTE key 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 5110 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 modem.

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

How Data is Accumulated

Counters accumulate the following information:

1. Total number of bits sampled
2. Number of synchronization errors
3. Number of sample errors for each sample position for both mark and space bits (Figure 3-2)
4. Number of consecutive error-free bits
5. 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, the system checks each sample for correctness by comparing the sample 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 bit-center sample is incorrect, the value in 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.

Status Messages

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

DS NOT READY – Ready status was not received from the modem because:

- The modem is not connected to the telephone line.
- The modem is not in the data mode.
- The modem cable is not connected to the 5110.
- The 5110 is not generating the 'data term ready' signal properly.
- The modem is not operating properly.

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

- The carrier from the distant modem is not detected.
- The 5110 is not providing the 'request to send' signal properly.
- The modem 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 '+received data' line and analyzes the samples
- Synchronizes sampling every 20 bits

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

Error Messages

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 ready' or 'clear to send' signal or both.

Graph Displays

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

- Space errors by sample
- Mark errors by sample
- Error-free 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 that were sampled on the '+received 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 five-digit decimal number in the total column at the extreme right of the graph.

The scale of errors displayed on 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 to 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 goes only 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 bps or 300 bps).

```
MDI 895         STEP NO. 026   ->
                    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 895         STEP NO. 027   ->
                     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 3-1) 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 count of the error-free bits that the 5110 received between failures.

Bit string lengths are divided into eight ranges. Each occurrence 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 five-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.

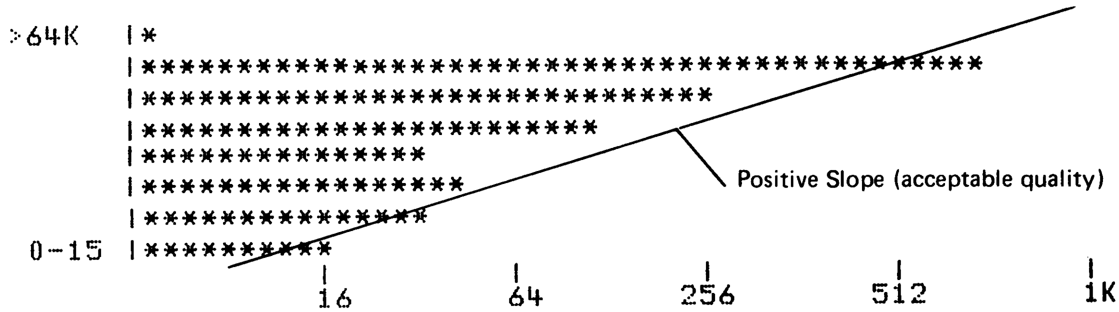
The scale of occurrences displayed on this graph is not linear. The value of each space on the graph increases with each 10-space division from left to right. For example, on the extreme left of the scale, every two asterisks in a bar represent approximately three occurrences of a bit string. In the 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 an error-free 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 network analysis subroutine since the last bit error or since the beginning of the test if no bit errors were detected.

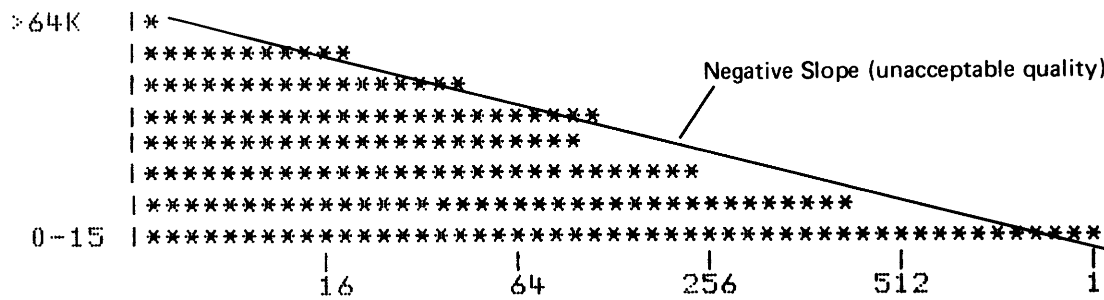
```
MDI 895          STEP NO. 026    ->
RANGE (BITS)          ERROR-FREE BIT STRINGS                              TOTAL
 >64K   |*                                                                00002
16K-64K |***********                                                      00020
 4K-16K |**********************                                           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
```

You can make a determination as to the quality of the line by averaging the graphic representation of the occurrence of error-free bit strings as shown below.




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




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

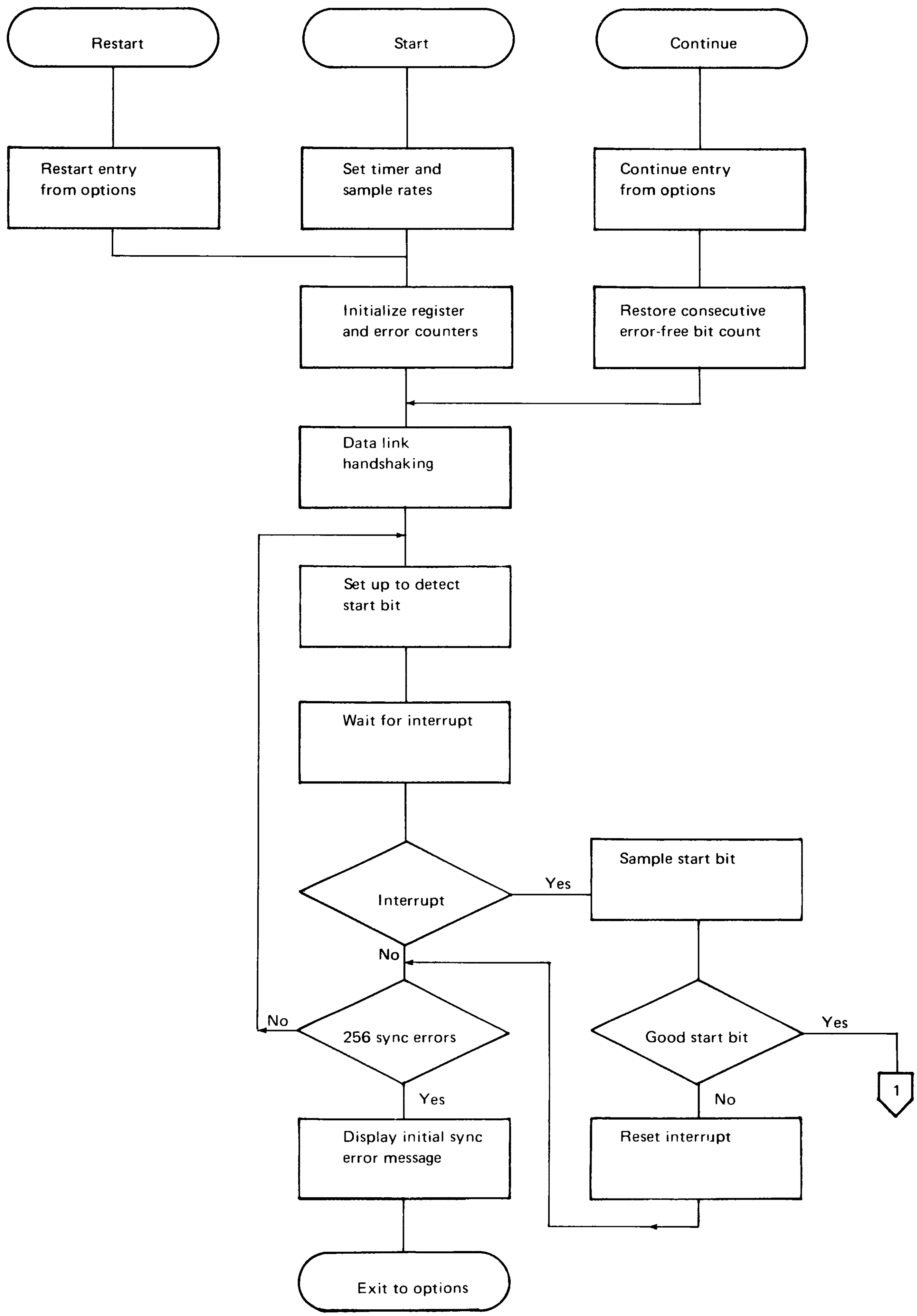
A negative slope indicates poor line fidelity, and a positive slope indicates more acceptable line quality.

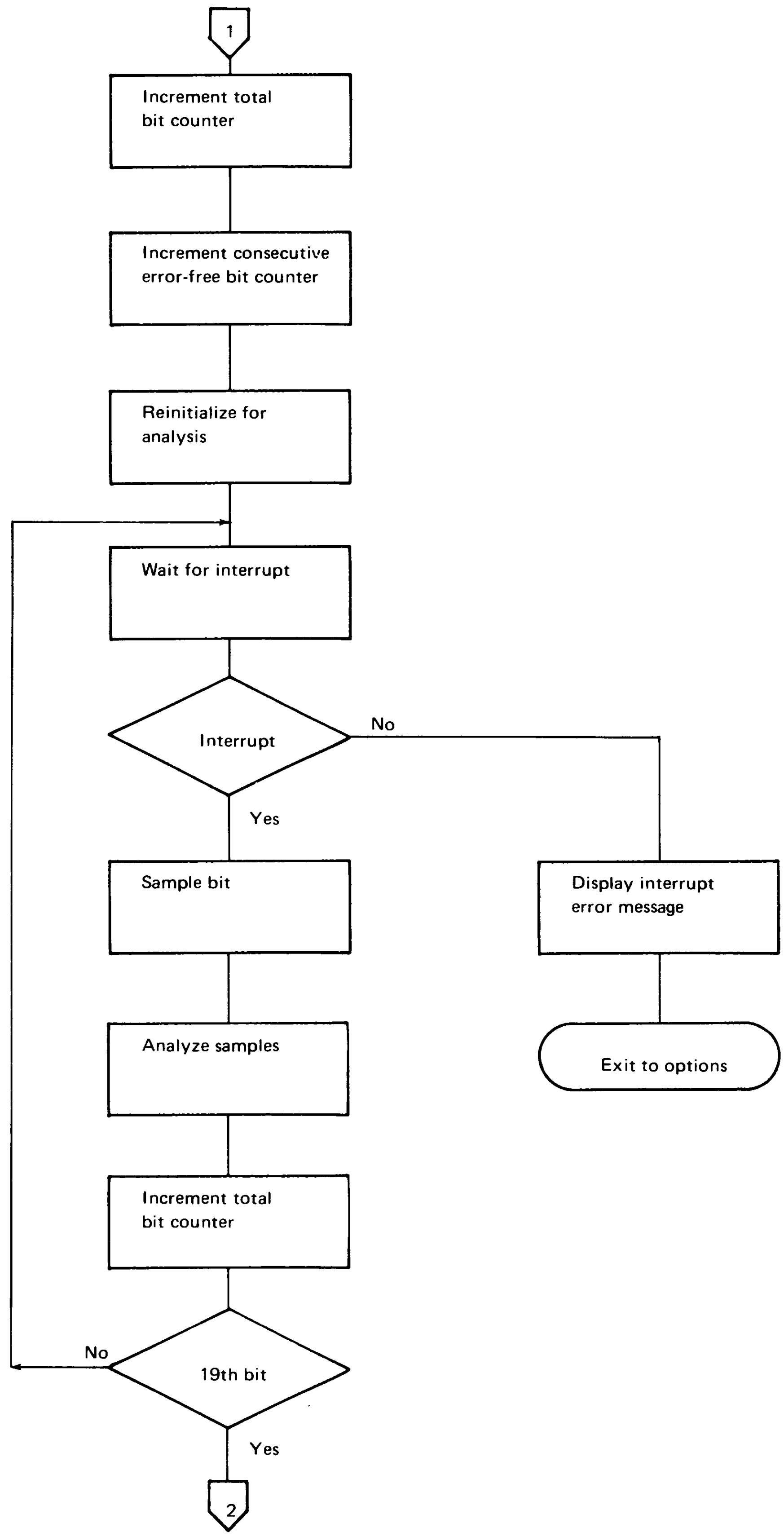
After viewing the error message and/or each page of results, you can select one of four options:

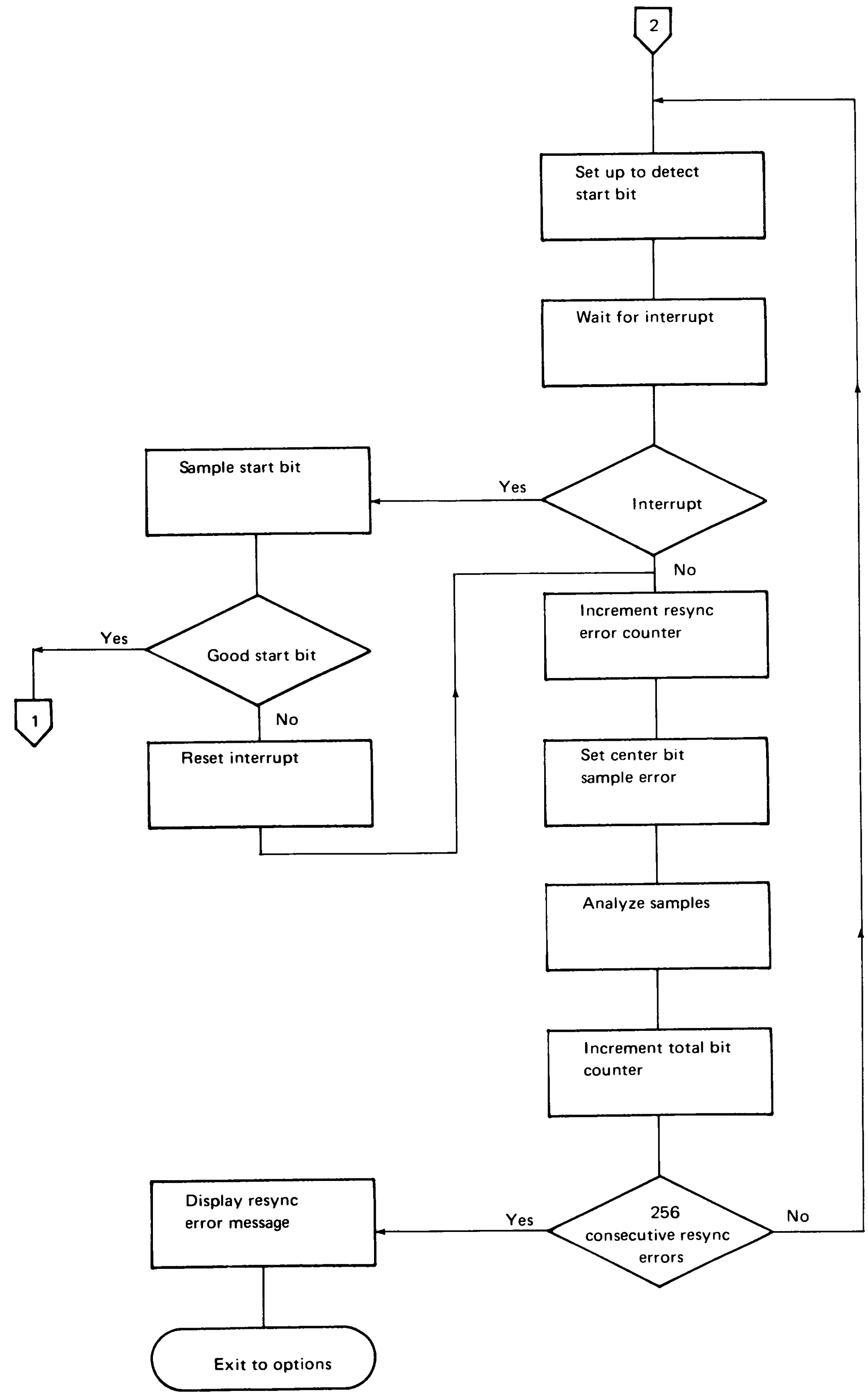
- Page to the next graph. Paging from the last graph returns to the first graph.
- Continue the test with all the error counters unaltered.
- Restart the test with all the error counters reinitialized.
- Terminate the test.

Network Analysis Subroutine Flowchart

The flowcharts on the following three pages represent the logic of the network analysis subroutine.


Restart
Start
Continue
Restart entry from options
Set timer and sample rates
Continue entry from options
Initialize register and error counters
Restore consecutive error-free bit count
Data link handshaking
Set up to detect start bit
Wait for interrupt
Interrupt
Yes
Sample start bit
No
256 sync errors
No
Good start bit
Yes
1
Yes
No
Display initial sync error message
Reset interrupt
Exit to options


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
Analyze samples
Exit to options
Increment total bit counter
No
19th bit
Yes
2



2
Set up to detect start bit
Wait for interrupt
Interrupt
Yes
No
Sample start bit
Good start bit
Yes
No
1
Reset interrupt
Increment resync error counter
Set center bit sample error
Analyze samples
Increment total bit counter
256 consecutive resync errors
Yes
No
Display resync error message
Exit to options

COMMUNICATIONS TROUBLESHOOTING GUIDE

Use the following chart as a guide when troubleshooting Asynchronous Communications feature problems:

- Run MDI 820; the asynchronous communications/serial I/O diagnostic MDI 820 tests the asynchronous communications hardware (asynchronous comm/serial I/O card, internal cables, and external cable).

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
  - Return the system to the customer.
  - Go to entry 002.

002

- Run the system diagnostics. Because the Asynchronous Communications feature uses existing ROS microprograms and a microprogram in read/write storage, you should run the system diagnostics to eliminate possible problems in these areas.

DID THE 5110 DIAGNOSTICS RUN ERROR FREE?

Y N
- Correct the problem indicated by the system diagnostics and rerun the user's job. If you are unable to correct the problem indicated by the system diagnostics, use existing support procedures.

  DOES THE USER'S JOB STILL FAIL?
  Y N
  - Return the system to the customer.
  - Go back to entry 002.

A

A

IS THE 5110 USING SWITCHED LINES?
Y N
- Display the 5110 communications diagnostic buffer to verify that the data is being transmitted and received by the 5110 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 procedures.
- If you dial the remote system again (two or three times if necessary, and wait 15 minutes after each 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 communications network. If the problem persists, have the user contact the telephone company.
- 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 5110 communications diagnostic buffer.

  IS THE DATA CORRECT IN THE BUFFER?
  Y N
  - The problem appears to be in the 5110. Use existing troubleshooting and support procedures.
  - The 5110 appears to be transmitting correctly. Retry the user's job; if it still fails, ask the user to contact the telephone company and request a line test. If the job still fails, use existing support procedures.

B

B

- Have the remote system send a test message to the 5110.

DID THE 5110 RECEIVE THE TEST MESSAGE CORRECTLY?

Y N

- Use the 5110 communications diagnostic buffer to check the received data.

  IS THE DATA IN THE BUFFER CORRECT?

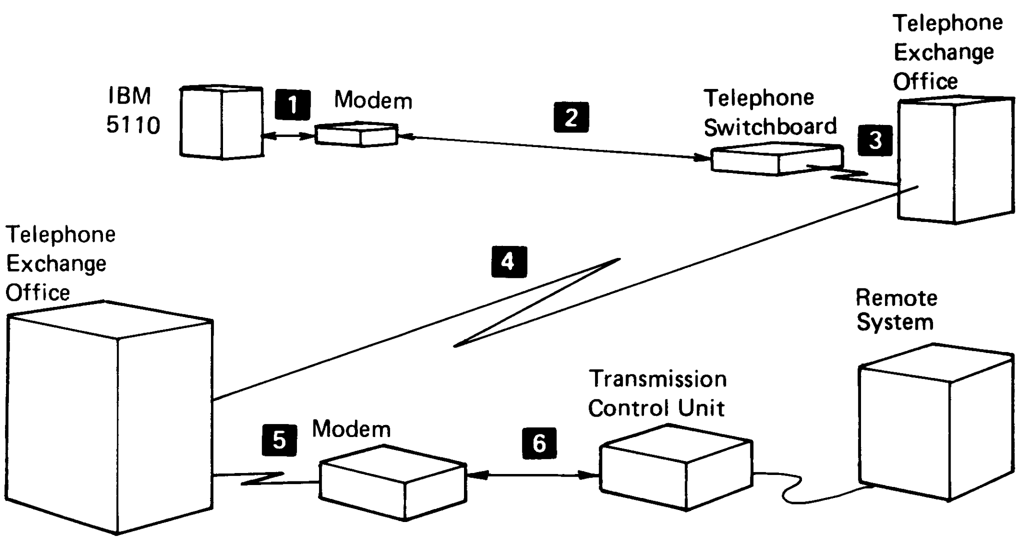
  Y N

  - 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 5110. Use existing troubleshooting and support procedures.

- The 5110 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 5110 and a remote system. Because communications network configurations vary from network to network, this diagram may not show the network configuration involving your system.



In this network, the data flow between the IBM 5110 and the remote system is as follows:

1. The 5110 sends the data to the modem.

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

3. The telephone switchboard sends the data to the 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 modem at the location of the remote system.

6. The modem 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 5110 and a remote system, you must dial the remote system location.

Switched lines may take different routes each time you dial a certain location. Such variations are caused by momentary 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 may 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 communications feature:

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 modem 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. (See *modem* under *Common Terms.*)

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

Full duplex indicates that data can be transmitted in both directions simultaneously. The 5110 Asynchronous Communications feature requires a full duplex facility.

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 Operations.*)

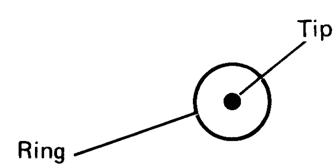
Modem is a device that supplies the carrier during data communications, modulates the carrier with the data signals, and transmits it over the communications lines. The modem also receives the modulated carrier from the communications lines, demodulates it (removes the data from the carrier), and sends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

Off hook indicates that the line is connected or active.

On hook indicates that the line is disconnected or inactive.

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

Example:



Space indicates no data bit on a data circuit.

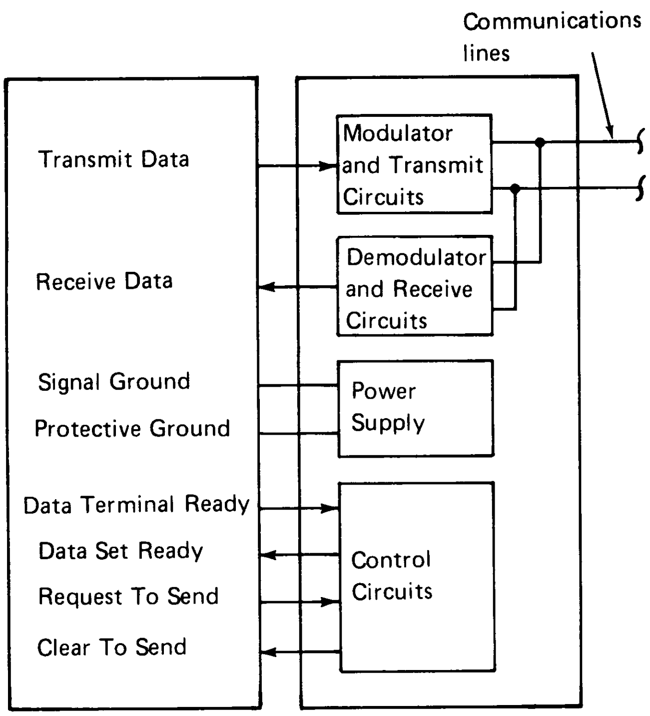
Start-stop is a term for asynchronous data transmission using a start bit and a stop bit to synchronize each character on the data circuit. (See *Data Format* in *Theory of Operation.*)

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

[1]A trademark of the Bell system.

MODEM FUNCTIONS

The modem converts the data signals from a terminal or system to signals that can be transmitted over the communications 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 modem 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 effects 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 modem 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 Modem EIA Interface* for an explanation of the EIA signals used with the 5110.)

The modem sequence of operations when the 5110 is transmitting data is as follows:

1. The 5110 turns on 'data terminal ready' to indicate that the 5110 is ready to communicate.

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

3. The 5110 turns on 'request to send' to indicate that the 5110 is ready to transmit data.

4. The modem turns on 'clear to send' to indicate that the modem is ready to transmit data and turns on the carrier. When the 5110 is not sending data, the carrier is held at a mark frequency.

5. The 5110 sends the data on the 'transmit data' line to the modem.

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

When the 5110 is receiving data:

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

2. The modem 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 5110 on the 'receive data' line. The first bit of a character (start bit) is a space level (+ voltage) on the 'receive data' line that indicates the start of a character to the 5110.

After all data is transmitted or received, the control lines (except 'data terminal ready' and 'request to send') and carrier are turned off until the 5110 requests another transmit or receive operation.

Appendix A. Character Code Charts

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 the meaning of the graphics that appear in the diagnostic buffer.

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

Character in Communications Diagnostic Buffer	Bit Value B	A	8	4	2	1	C	Meaning	
j			8	4	2			**Up-shift**	These characters can be received and transmitted.
ɩ			8	4	2	1	C	Ⓒ	
o	B		8	4		1	C	New line	
≠	B		8	4	2		C	Backspace	
∈	B	A	8	4		1		Tab	
⌷	B	A	8	4	2			**Down-shift**	
(EBCD) 9(Correspondence Code)			8		2	1		Ⓓ	
							C	Space	
h			8	4			C	Punch on	These characters are received, but are ignored. They cannot be transmitted.
≤			8	4		1		Reader stop	
L		A	8	4	2		C	EOB	
→		A	8	4	2	1		Prefix	
i	B	A	8	4			C	Punch off	
g	B	A	8	4	2	1	C	**DEL**	
d		A	8		2			Invalid idle	
e	B		8		2			Invalid idle	
f	B	A	8		2		C	Invalid idle	
c	B		8	4	2	1		Valid idle	
↑		A	8	4				Bypass	These characters are received and used. They cannot be transmitted.
b		A	8	4		1	C	Line feed	
↓	B		8	4				Restore	
$\tilde{\wedge}$								Receive parity error	
$\tilde{\vee}$								Receive stop bit error	
$\underline{\phi}$								Indicates a transmit long space	

Code Chart for EBCD Data Characters

GRAPHIC CHARACTERS						EBCD BIT VALUE						
APL		BASIC		EBCD								
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	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	⍴	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	⍵	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
]	)	#	¨	#	¨			8		2	1	
←	→	@	ç	@	ç		A					
x	÷	&	+	&	+	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 display. The] is also used to represent a Ⓓ in the communications diagnostic buffer display.

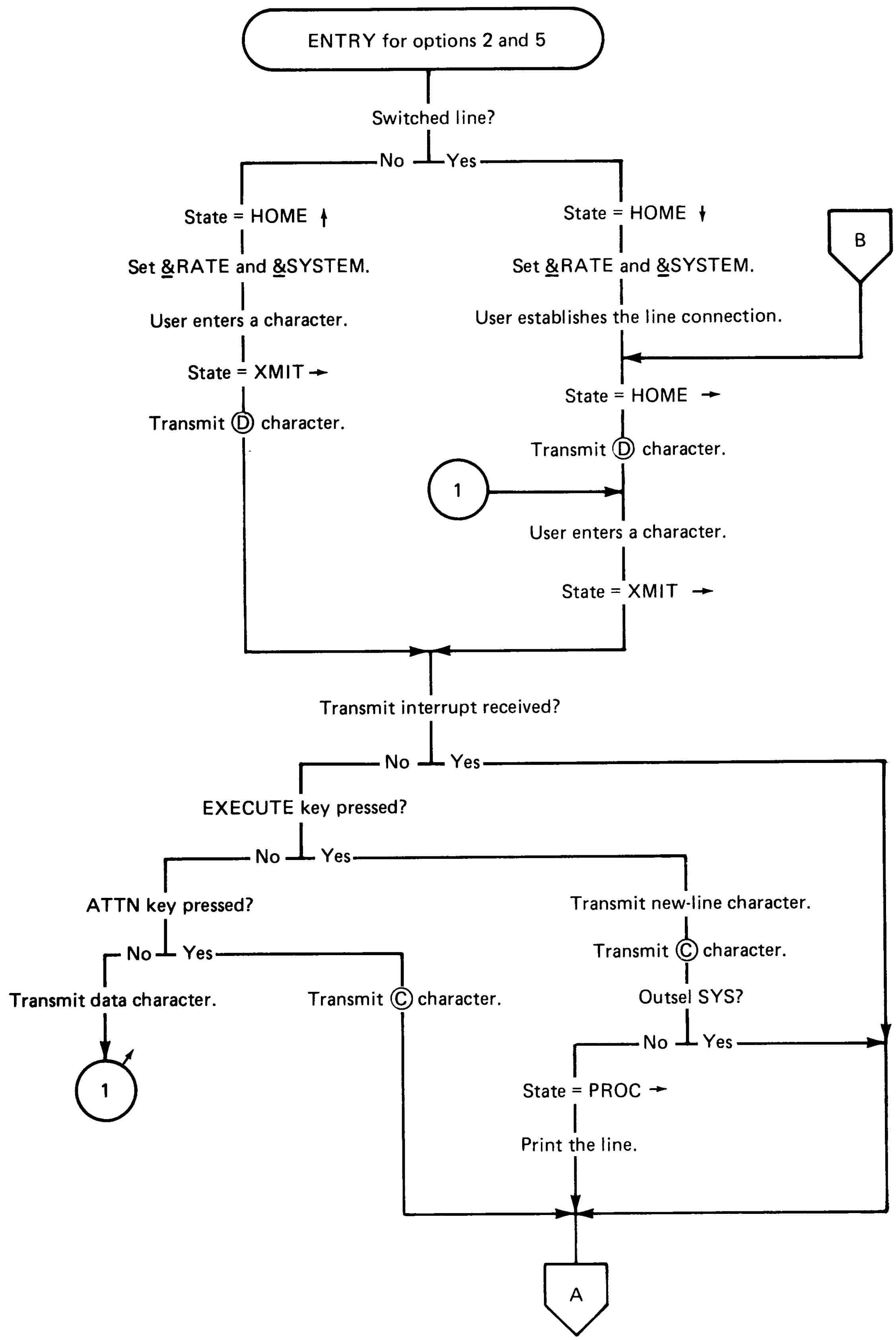
Code Chart for Correspondence Code Data Characters

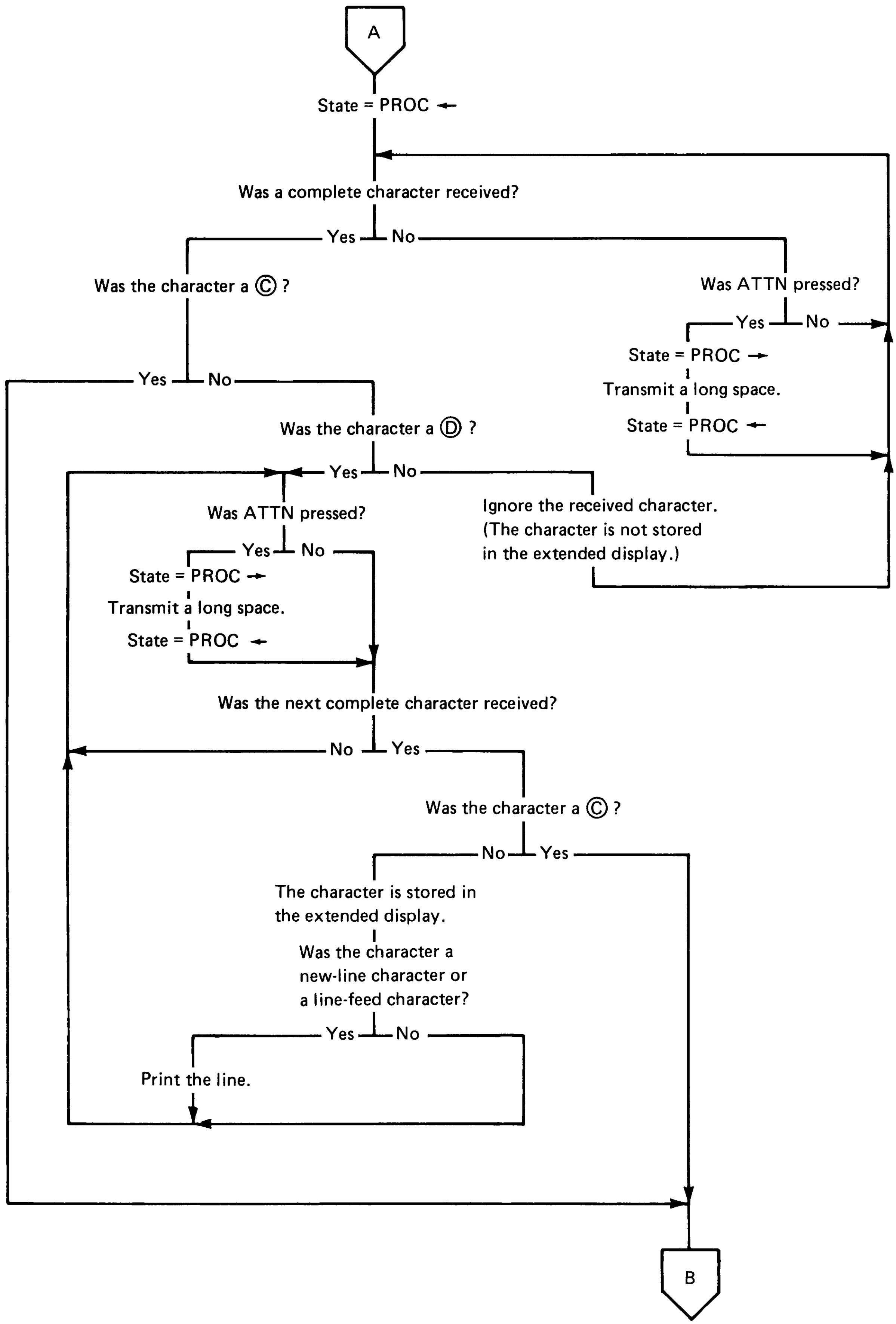
GRAPHIC CHARACTERS												
APL		BASIC		CORRESPONDENCE		CORRESPONDENCE BIT VALUE						
LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	LOWER SHIFT	UPPER SHIFT	B	A	8	4	2	1	C
A	⍺	a	A	a	A	B			4	2	1	C
B	⊥	b	B	b	B		A	8		2	1	C
C	∩	c	C	c	C		A		4	2	1	C
D	⌊	d	D	d	D		A		4		1	
E	∊	e	E	e	E		A		4			C
F	_	f	F	f	F	B	A			2	1	C
G	∇	g	G	g	G	B	A				1	
H	∆	h	H	h	H		A	8			1	
I	⍳	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	⍵	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

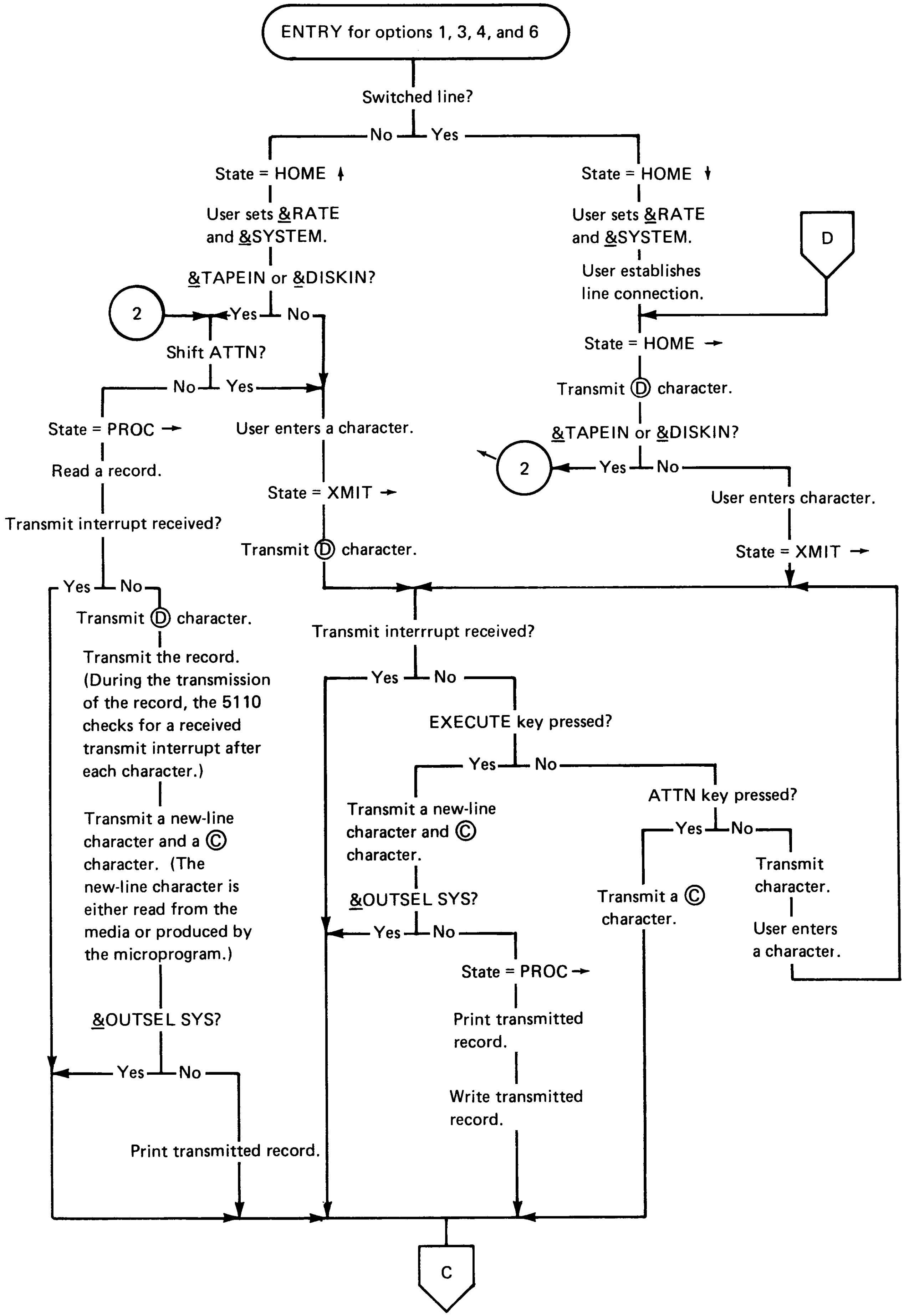
These characters appear in the communications diagnostic buffer display. The 9 is also used to represent a Ⓓ in the communications diagnostic buffer display.

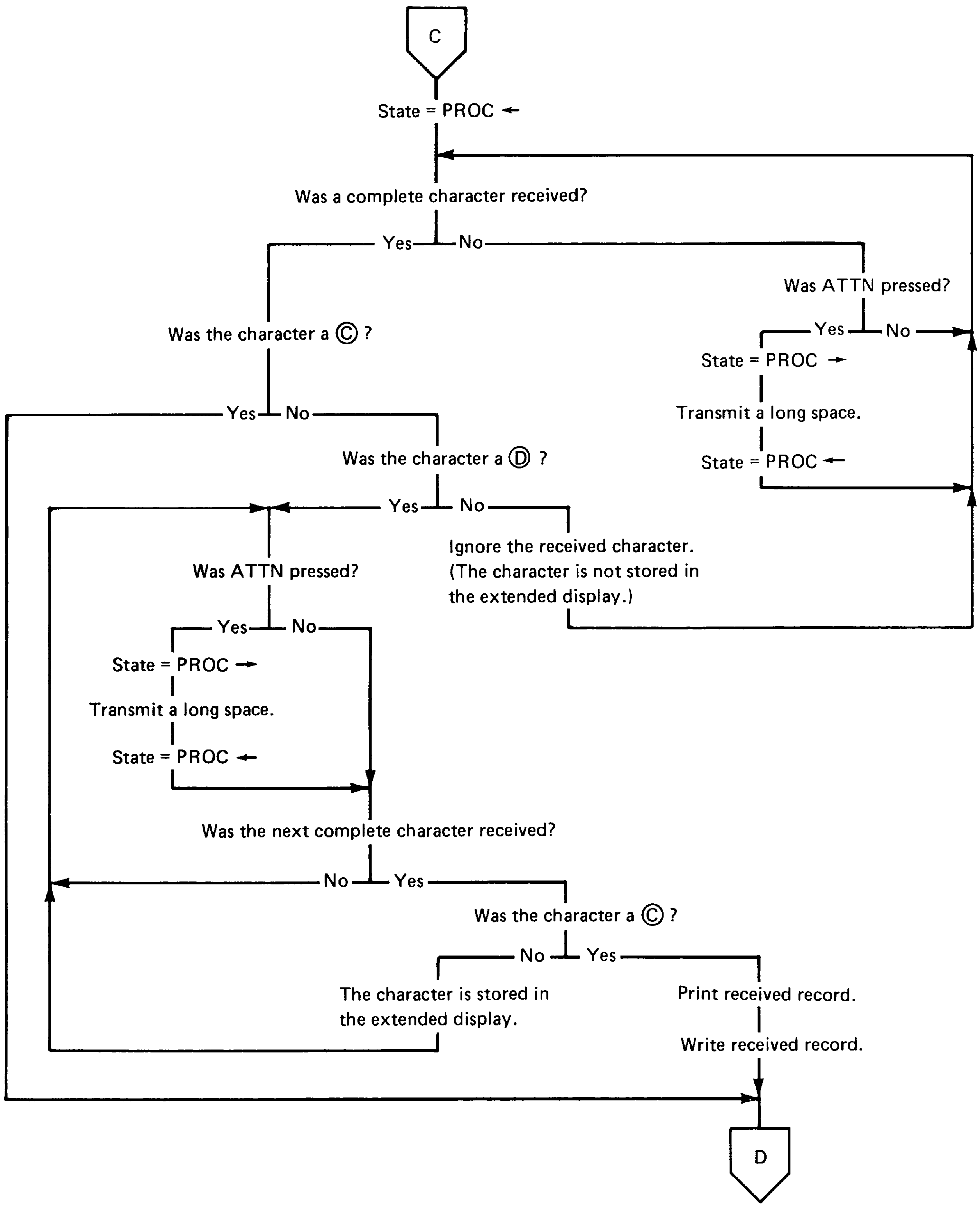
Appendix B. Asynchronous Communications Sequence Charts

The following charts represent the action sequences during normal transmit and receive operations with the 5110 Asynchronous Communications feature. These charts assume that the printer is attached and is prepared for operation, that the tape or diskette devices are prepared, and that appropriate files are open for I/O operations.


ENTRY for options 2 and 5
Switched line?
No
Yes
State = HOME
Set &RATE and &SYSTEM.
User enters a character.
State = XMIT
Transmit Ⓓ character.
State = HOME
Set &RATE and &SYSTEM.
User establishes the line connection.
B
State = HOME
Transmit Ⓓ character.
1
User enters a character.
State = XMIT
Transmit interrupt received?
No
Yes
EXECUTE key pressed?
No
Yes
ATTN key pressed?
No
Yes
Transmit data character.
1
Transmit Ⓒ character.
Transmit new-line character.
Transmit Ⓒ character.
Outsel SYS?
No
Yes
State = PROC
Print the line.
A


A
State = PROC
Was a complete character received?
Yes
No
Was the character a Ⓒ ?
Was ATTN pressed?
Yes
No
State = PROC
Transmit a long space.
State = PROC
Yes
No
Was the character a Ⓓ ?
Yes
No
Was ATTN pressed?
Ignore the received character.
(The character is not stored
in the extended display.)
Yes
No
State = PROC
Transmit a long space.
State = PROC
Was the next complete character received?
No
Yes
Was the character a Ⓒ ?
No
Yes
The character is stored in
the extended display.
Was the character a
new-line character or
a line-feed character?
Yes
No
Print the line.
B


ENTRY for options 1, 3, 4, and 6
Switched line?
No
Yes
State = HOME
User sets &RATE and &SYSTEM.
&TAPEIN or &DISKIN?
2
Yes
No
Shift ATTN?
No
Yes
State = PROC
Read a record.
Transmit interrupt received?
Yes
No
Transmit Ⓓ character.
Transmit the record. (During the transmission of the record, the 5110 checks for a received transmit interrupt after each character.)
Transmit a new-line character and a Ⓒ character. (The new-line character is either read from the media or produced by the microprogram.)
&OUTSEL SYS?
Yes
No
Print transmitted record.
User enters a character.
State = XMIT
Transmit Ⓓ character.
State = HOME
User sets &RATE and &SYSTEM.
User establishes line connection.
D
State = HOME
Transmit Ⓓ character.
&TAPEIN or &DISKIN?
2
Yes
No
User enters character.
State = XMIT
Transmit interrrupt received?
Yes
No
EXECUTE key pressed?
Yes
No
Transmit a new-line character and Ⓒ character.
&OUTSEL SYS?
Yes
No
State = PROC
Print transmitted record.
Write transmitted record.
ATTN key pressed?
Yes
No
Transmit a Ⓒ character.
Transmit character.
User enters a character.
C


C
State = PROC ←
Was a complete character received?
Yes
No
Was ATTN pressed?
Yes
No
State = PROC →
Transmit a long space.
State = PROC ←
Was the character a Ⓒ ?
Yes
No
Was the character a Ⓓ ?
Yes
No
Ignore the received character. (The character is not stored in the extended display.)
Was ATTN pressed?
Yes
No
State = PROC →
Transmit a long space.
State = PROC ←
Was the next complete character received?
No
Yes
Was the character a Ⓒ ?
No
Yes
The character is stored in the extended display.
Print received record.
Write received record.
D

Index

READER'S COMMENT FORM

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

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

Page Number *Error*

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

Page Number *Comment*

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

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

Name ______________________

Address ______________________

____

Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

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.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

READER'S COMMENT FORM

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

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

Page Number *Error*

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

Page Number *Comment*

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

Name ____________________

Address ____________________

____

- **No postage necessary if mailed in the U.S.A.**

Cut Along Line

Fold Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

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.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM 5110 Asynchronous Communications Feature Maintenance Information Manual Printed in U.S.A. SY31-0557-0

SY31-0557-0

IBM 5110
System Maintenance
Library