

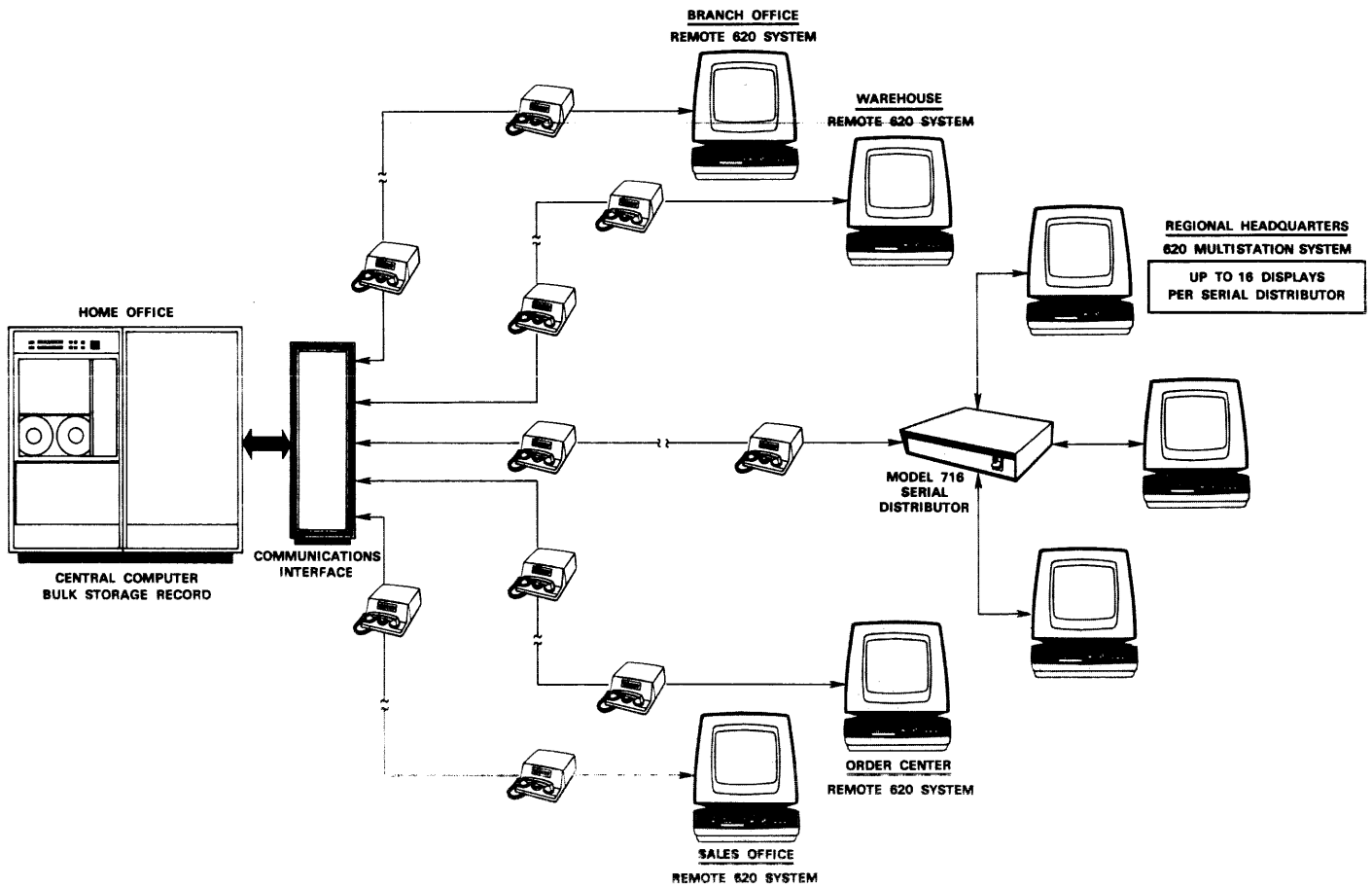
SANDERS 620*

STAND-ALONE DATA DISPLAY SYSTEM

MANUAL



Sanders 620 Display is prepared to meet your total system needs . . . for stand-alone terminals . . . a cluster of terminals . . . or both . . . in a local or nationwide network of instant computer communications



MODEL 620
STAND-ALONE
DATA DISPLAY SYSTEM
MANUAL

FEBRUARY 1969
REVISION 2



DATA SYSTEMS DIVISION

D.W. HIGHWAY SOUTH, NASHUA, N.H. 03060

THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION OF SANDERS ASSOCIATES, INC., AND IS TO BE USED ONLY FOR THE PURPOSE FOR WHICH IT HAS BEEN SUPPLIED. THIS DOCUMENT IS NOT TO BE DUPLICATED IN WHOLE OR IN PART WITHOUT PRIOR WRITTEN PERMISSION FROM A DULY AUTHORIZED REPRESENTATIVE OF SANDERS ASSOCIATES, INC.

SA-00-614 (8-67)

INTRODUCTION

This manual contains a detailed component description of the Sanders' 620 Data Display System as well as programming considerations for the 620 with IBM System/360. The information is to be regarded as proprietary and is to be used only for the purpose for which it has been supplied. This document is not to be duplicated in whole or in part without prior written permission from a duly authorized representative of Sanders Associates, Inc.

TABLE OF CONTENTS

Section 1

Functional Description	1
----------------------------------	---

Section 2

System Configuration	2
I/O Section	2
Edit Section	2
Memory Access Section	3
Memory Section	3
Display	4
Power Supply	4
Keyboard	4

Section 3

System Operation	5
Input/Output Operation	5
General	5
Synchronous Operation	5
Asynchronous Operation	6
Communication Sequences	6
Non-Response Mode	7
Response Mode	9
Error Checking	12
Edit Operation	13
Format Type Mode	13

TABLE OF CONTENTS (Cont)

Type Mode 14
 Conversation Type Mode 17
 Description of Hard Copy Option 18
 Hard Copy Option Components 18
 Hard Copy Buffer I 18
 Printer 19
 Transmission Cable 19
 Keyboard 19
 Hard Copy Functional Operation 19
 Hard Copy Code Conversion 19
 Hard Copy Modes of Operation 20

Section 4

Programming Considerations for Use of 620 with IBM System/360 21
 Remote Transmission - (2400 Bits per second or less) 21
 OS BTAM - (Remote Connection) 21
 OS QTAM - (Remote Connection) 22
 DOS BTAM - (Remote Connection) 23
 DOS QTAM - (Remote Connection) 24
 Device Dependent Module (DDM) Description 25
 Detailed Message Structures - Remote Operation 25
 Access-Method Dependent Information 26
 Using a DDM 26
 Message Processing Macros 27
 EDITMOVE Macro-Instruction 27
 TABLES Macro-Instruction 28

Section 5

Timing Considerations for 620 Control Operations 29
 Format Type Mode 30
 Type Mode 30
 Conversation Type Mode 31
 Table of f Values 32

TABLE OF CONTENTS (Cont)

Section 6

Interface Requirements and Characteristics	33
Hardware Interface	33
Data Set Interface	33
Serial Distributor Interface	33
Software Interface with IBM System/360	34
Special OS System Generation Considerations	34
BTAM and QTAM-Remote Connection	34
Caution	34
Special DOS System Generation Considerations	34
BTAM and QTAM-Remote Connection	34
Caution	35

Section 7

Installation Planning Data	36
Input Power Requirements	36
Physical Specifications	36
Display Terminal	36
Dimensions and Weight	36
Connectors	36
Cables	36
Keyboard	37
Dimensions and Weight	37
Connectors	37
Function Keys	37
Environmental Conditions	37
Operating	37
Storage	37

APPENDIX

A. 620 Control - Character Abbreviations 38

B. 620 CODE TABLE - (USASCII SUBSET) 39

C. 620 Keyboard 40

D. 620 Character-Generator Racetrack Schematic 41

E. Table of 620 Control Operations 42

F. Comparison Table - 620/720 Message Formats 43

G. Translation Tables - (ASCII-8 to EBCDIC and EBCDIC to ASCII-8) 44

H. Glossary of Terms 46

SECTION 1

FUNCTIONAL DESCRIPTION

The 620 is a desk-top stand-alone alphanumeric CRT display. It has a 780-character programmable memory and the optional capability of writing into any one of over 2,000 screen locations. Optional edit features include Horizontal and Vertical tabs, fixed and variable data blocks, and conversation mode. All I/O, editing, memory, display and power supply components are self-contained in a single terminal package. It is designed to operate in a remote environment via data set modem connection to a digital Central Processor.

The 620 keyboard contains alphanumeric and control function keys as defined by Appendix C. Data entry via the keyboard causes screen character generation or function control to occur simultaneously with each keystroke.

Hardcopy may be added to the 620 with the use of the hardcopy adapter and a teleprinter. The remote user may initiate the copy of either variable block information or the entire screen content by depressing the function keys associated with those operations. The computer may poll and initiate a printing sequence by inserting a print command instruction prior to the ETX of its outbound message.

Data transmission may be on a polled or a contention basis and can be either synchronous or asynchronous. A subset of Standard USASCII Code is used for all data transmission and the data set interconnection meets EIA standard RS 232 B Specifications.

SECTION 2

SYSTEM CONFIGURATION

The 620 System consists of seven functional blocks. They are: 1. I/O Section, 2. Edit Section, 3. Memory Access Section, 4. Memory Section, 5. Display Section, 6. Power Supply, and 7. Keyboard.

I/O SECTION

The function of the I/O Section is to control the transfer of data between a data set and the 620 System. The I/O Section performs the following functions:

- a. Generates character parity and LRC on the outbound messages.
- b. Checks for parity error and LRC error on incoming messages.
- c. Decodes the header sequence and recognizes a specific address for inbound messages.
- d. Encodes the header sequence for outbound messages.
- e. Provides logic level conversion and synchronization with the Data Set.

EDIT SECTION

The function of the Edit Section is to perform all editing operations for data received from either the keyboard or the computer, and for all data sent to the computer.

The basic edit function provides the capability of entering data on the screen from left to right and top to bottom through 780 locations. All entry is in format type mode.

X

Optional Edit features include:

- Horizontal and Vertical Tab -- where a HT causes the cursor to move 4 screen spaces to the right of the starting location and inserts 1 HT character on the delay line, and a VT causes the cursor to move 4 lines down (to the beginning of the new line) on the screen and inserts 1 VT character on the delay line. All 2,000 +screen locations are accessible with this option.
- Conversation Mode -- where each alternate Computer/User entry is automatically positioned at the beginning of a successive line. For example, the user may enter data on line 1, the computer automatically uses line 2 for its reply; the user then is positioned to line 3 for his next entry, the computer then replies on line 4, and so on.
- Format Mode -- where start delta \triangleright and stop delta \triangleleft are used to define variable data-entry fields. Under format control, the user is automatically prohibited from writing into any fields other than those defined by the start and stop delta characters. The completion of one field causes the cursor to automatically move to the beginning of the next available field, (if there is one available), defined by deltas.

Not in use.

Format Mode keys are provided to allow the use of these Edit options. New formats are created in Format mode by programming/systems personnel and can then be used in Type Mode by the operators. HT and VT are considered Format Mode keys.

MEMORY ACCESS SECTION

The function of the memory access section is to provide the interface between the 620 Memory and the Edit and/or Display sections.

MEMORY SECTION

The function of the memory section is to store all alphanumeric and function-control characters required to format, edit, and generate a display. Either the total contents or the delta-defined data blocks of the memory may be dumped to the tele-communications line.

DISPLAY SECTION

The function of the display section is to provide a visual representation of the data stored in memory on the face of the CRT.

POWER SUPPLY

The function of the power supply is to provide the proper voltage and current requirements to each of the other operating sections.

KEYBOARD

The function of the keyboard is to enable the operator to enter, display, and transmit data through the 620 System. The keyboard is provided with electrical interlock such that the first key-depressed will be entered into memory. A two key roll-off feature is provided so that if two keys are depressed, the second key will not be entered into memory until the first key is released.

There are 42 alphanumeric and special symbol keys, 19 control keys and a space bar. One of the control keys is a repeat key which, when depressed along with a second key, will make repeated entries at a 30 cps rate. Cycle left and Cycle right keys are also provided to aid rapid cursor movement through screen locations.

SECTION 3 SYSTEM OPERATION

Note: Available 620 Screen matrices include:

64 characters/line by 32 lines (Horizontal Screen)

52 characters/line by 40 lines (Vertical Screen)

84 characters/line by 32 lines (Expanded Horizontal Screen)

These matrix options may be regarded as defining the screen record positions into which data may be written, and although the following information is based on a 64×32 matrix, the same operating considerations apply to the alternative matrices.

General

The system operation is described in three sections, (1) I/O Operation, (2) Edit Operation, and (3) Description of Hard Copy Option. Appendix B identifies the codes that are used in the 620 system.

INPUT/OUTPUT OPERATION

GENERAL

The 620 display operates as a remote terminal capable of transmission and reception of serial data at rates up to 2400 bits per second (bps). The 620 may be connected to a modem, or a number of 620's may be connected to a modem via a Sanders Associates, Inc. Model 716 Serial Distributor. The modem being used determines the data transfer rate and dictates whether data transfer is to be effected in the synchronous or asynchronous mode.

SYNCHRONOUS OPERATION

In the synchronous mode, all characters transferred contain eight bits. Seven of the eight bits in each character are data bits, and the eighth bit is a vertical parity bit. In this mode, transmit and receive clocks must be provided to the 620 by the modem. Each message sent to the 620 must be preceded by at least two successive ASCII SYNC characters. Each message sent by the 620 will, in turn, be preceded by from two to four successive SYNC characters, the exact number being a 620 option. Once synchronization

has been established, there can be no time gaps between bits or words. Synchronous communication takes place at a rate of 2000 or 2400 bits per second.

ASYNCHRONOUS OPERATION

In the Asynchronous mode, a ten or an eleven bit character is used. This character consists of seven data bits, one vertical parity bit, one start bit, and one or two stop bits, the exact number of stop bits being a 620 option. Asynchronous communication is also referred to as "start-stop" communication. SYNC characters are not used, but rather the start bit of each character is used for synchronization; therefore characters need not follow each other at a fixed succession rate because synchronization is reestablished with each start bit. In the asynchronous mode, the 620 uses an internal clock in transmission and reception of data. Data transmission rates from 110 to 2400 bits per second may be used. The transmission rate is established by selection of the appropriate jumper connections on the printed circuit I/O card.

COMMUNICATION SEQUENCES

Either of two basic communication sequences is available in the 620. There is a response sequence and a non-response sequence. These sequences are 620 options. The basic difference between them is that the response sequence requires use of longitudinal redundancy checks and message acknowledgements which are not required when the non-response sequence is used.

In both the response and non-response modes, there are two basic operations which may be performed. These are the poll and write operations. The poll operation is a request by the CPU for data from the 620 while the write involves transmission of data from the CPU to the 620.

In both the response and non-response modes, a poll or a write must be initiated by the CPU which transmits a four-character command sequence to the 620. This command sequence may also be referred to as an addressing sequence. The four characters of this sequence are as follows:

- a. SOH (Start of Header)
- b. CUA (620 address)
- c. X '09' (unused address when addressing 620)
- d. COMMAND (Write or Poll)

The CUA (Control Unit Address) for the 620 may be any of the 128 possible USASCII characters, but not SYNC or EOT. Each 620 is hard wired to accept one and only one such address. Any command sequence containing an address other than that associated with a given 620 will be ignored by that 620. This feature permits use of the 620 with the Model 716 Serial Data Distributor. If a command sequence is received by a number of displays, only the display whose address corresponds to the CUA of the command sequence will recognize the command.

The DUA (Display Unit Address) or second-level address of the command sequence is used primarily for compatibility with Sanders 720/360 message structures. The DUA in a command sequence sent to any 620 must be the 620 code for horizontal tab. (This is purely arbitrary and, of course, does not result in writing a horizontal tab on the display screen.) This code will also be the second level address sent by any 620 in transmission.

The command itself is the fourth character of the command sequence. Only two commands are recognized, the WRITE and the POLL. All other codes received at this point result in no operation.

There is one case in which an operation may begin without initiation of a command sequence by the CPU. If it is desired that a system operate on a contention (i.e., non-polling) basis, the 620 contention option may be enabled to allow the display terminal to transmit data upon operation of the "send" key on the keyboard. With this option enabled, transmission takes place without receipt of a poll from the CPU.

Non-Response Mode

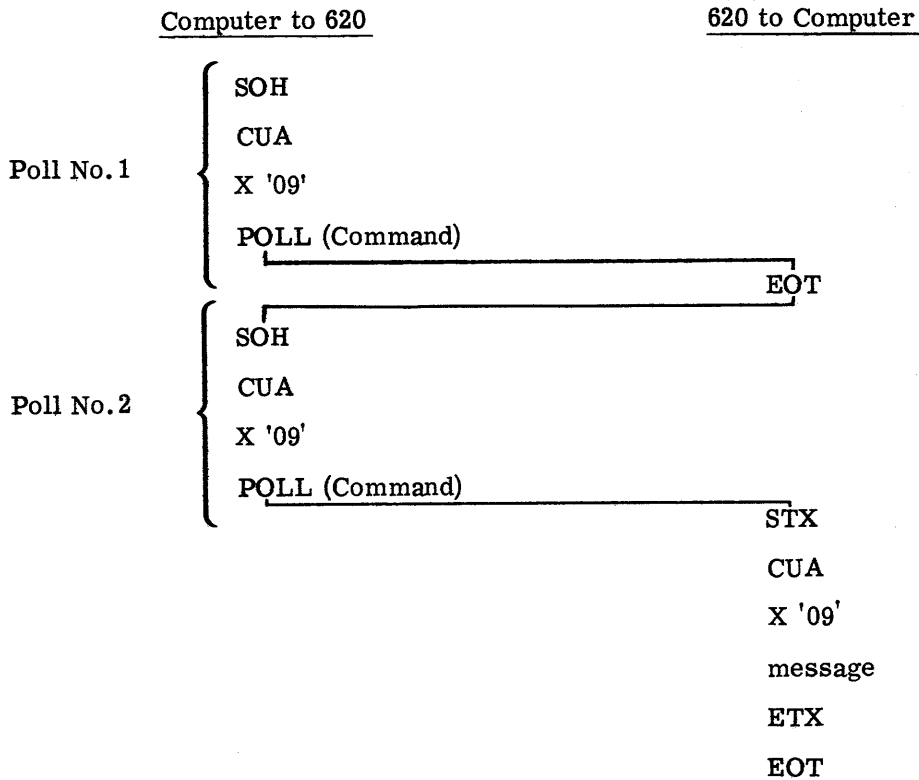
Write Operation -- In the non-response write operation, the CPU initiates the WRITE command sequence and follows this immediately with a start of text (STX) character. Following the STX from the CPU is the message which is terminated by an end of the text (ETX) character. The entire sequence is then terminated by an end of transmission (EOT) character from the CPU. The full sequence is as follows:

Computer to 620

```
SOH
CUA
X '09'
|
```

|
 WRITE (Command)
 STX
 Message
 ETX
 EOT

Poll Operation -- In the non-response poll operation, the CPU command sequence ends with the POLL character. At this point, if the 620 has no message to transmit, it responds with an EOT. If the 620 does have a message to transmit, it responds with STX, CUA, X '09', message, ETX, and EOT. The full sequence is shown in the table below. In the example shown, the 620 has no message at the end of the first poll, but does have a message at the end of the second poll.



Contention -- In contention, the 620 transmits upon depression of the keyboard SEND key. The sequence is shown in the following table.

620 to Computer
 STX
 CUA
 X '09'
 |

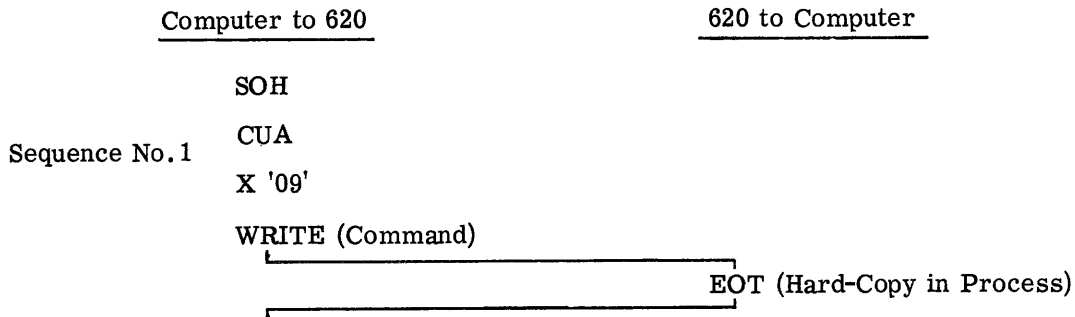
|
 Message
 ETX
 EOT

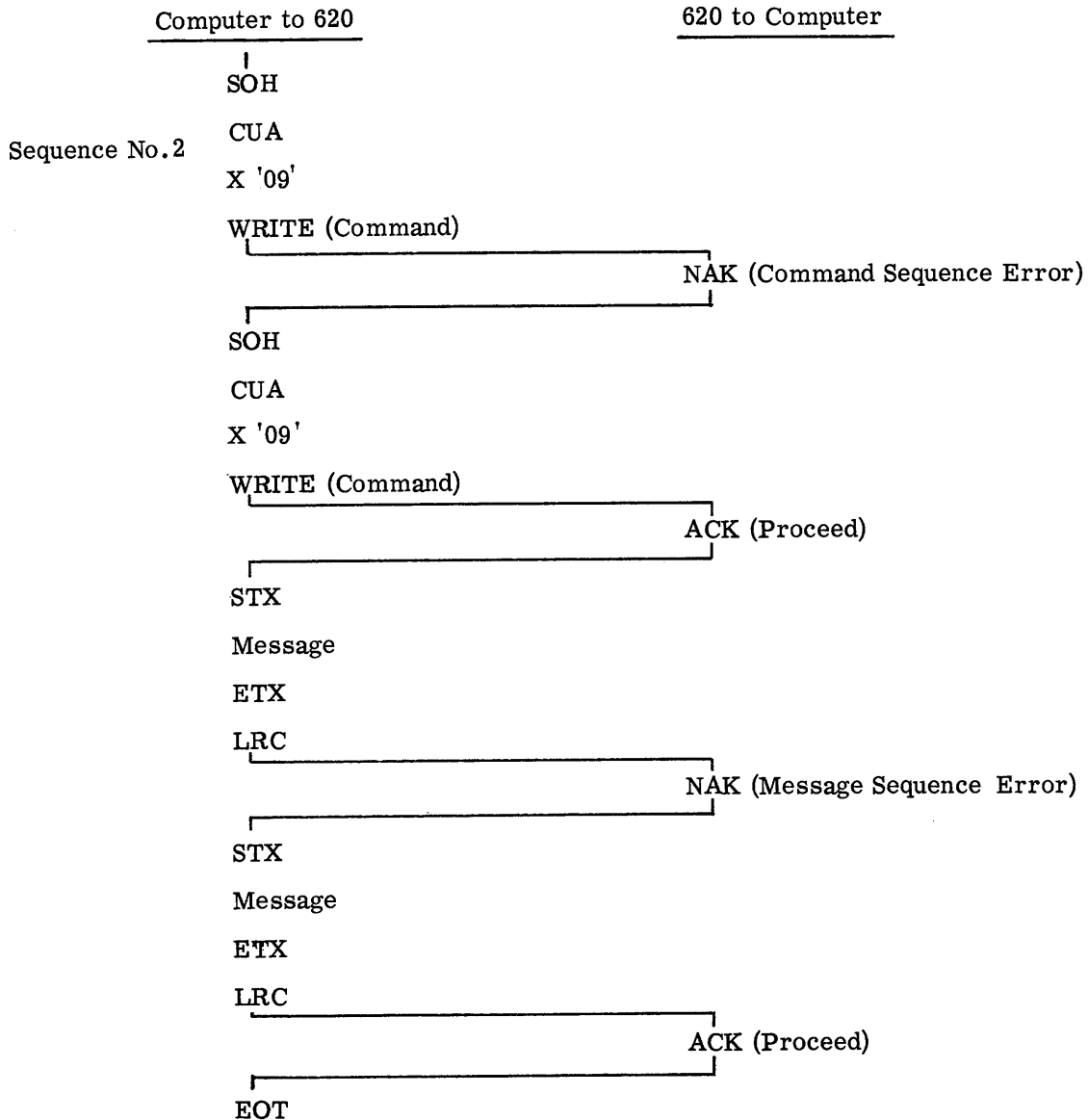
Response Mode

Write Operation -- The write operation in the response mode is similar to the write in the non-response mode. It differs, however, in certain respects. After receipt of the write command sequence, the 620 must respond to the CPU with a negative acknowledge (NAK) or a positive acknowledge (ACK) indicating detection or non-detection of a vertical parity error in the command sequence. If a NAK is transmitted by the 620, the CPU must, in turn, retransmit the command sequence. If an ACK is sent by the 620, the CPU then transmits the message sequence: STX, message, ETX, LRC. The LRC character is the longitudinal redundancy character accumulated by a longitudinal redundancy check over all characters from, but not including, STX up to and including ETX. If a longitudinal or vertical parity error is detected by the 620, it awaits completion of the message from the CPU and then responds to the CPU with a NAK. Otherwise, it responds after message completion with an ACK. The NAK signals the CPU to retransmit the message while the ACK signals the CPU to transmit EOT and end the write sequence.

If the 620 responds to the initial write command sequence with an EOT, it is an indication that a 620 hard copy (optional) operation is in process and that the message from the computer cannot be accepted. The computer must re-initiate the command sequence to execute the write.

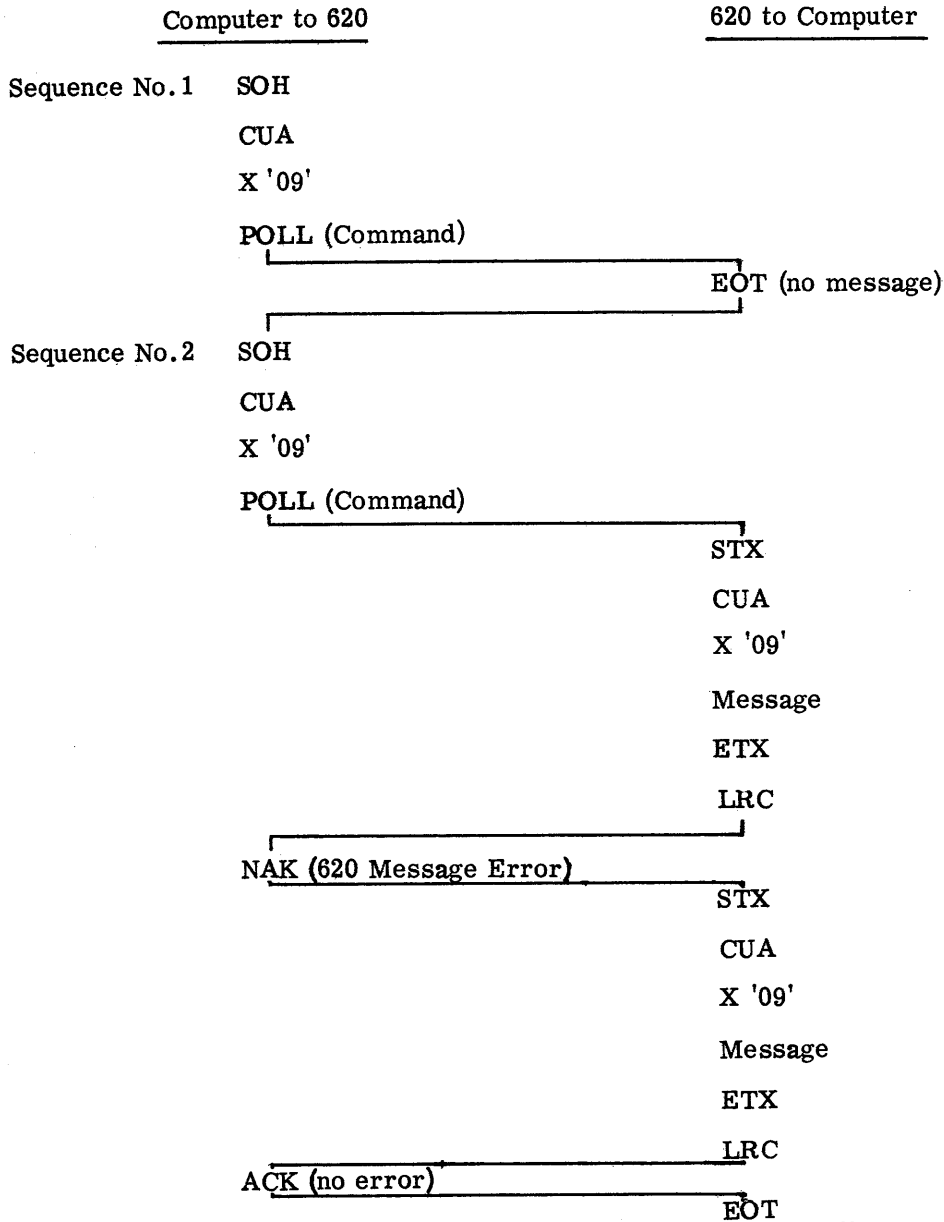
The table below shows the write sequence with the various 620 responses:



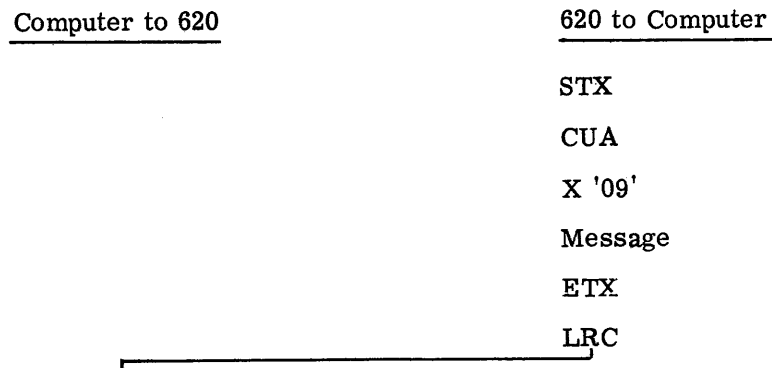


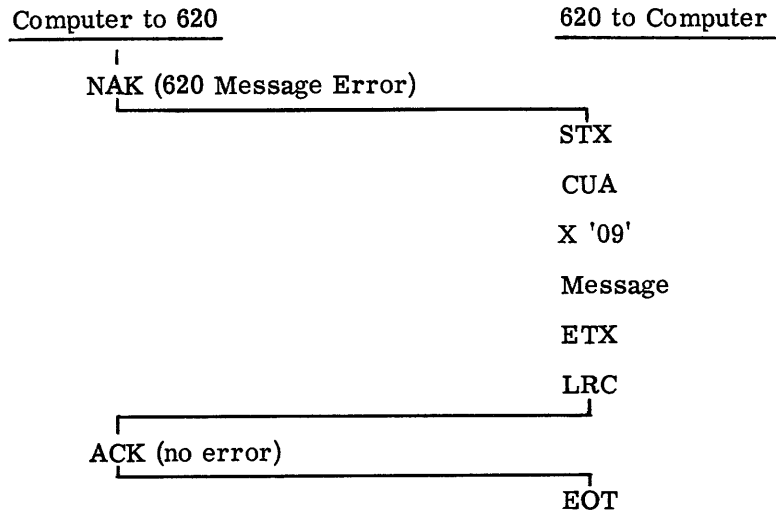
Poll Operation -- The poll command sequence for the response mode is the same as for the non-response mode. If the 620 has no message it replies to the poll with EOT. If the 620 has a message for the CPU it sends STX, CUA, X'09', Message, ETX, LRC. The LRC is accumulated from, but not including STX, up to and including ETX.

The CPU responds to the 620 Message with ACK or NAK according to the results of the longitudinal and vertical parity checks at the CPU interface. A NAK signals the 620 to retransmit the full message while ACK calls for 620 transmission of an EOT to terminate the entire sequence. The table below illustrates the response poll sequence:



Contention -- In contention, operation is the same as in polling except that the sequence is initiated by the 620 message transmission. Operation is shown in the table below.





ERROR CHECKING

In synchronous data communication, the parity bit of each data byte must be such that odd parity over the byte results. In asynchronous data communication, the parity bit of each byte must be such that even parity over the seven data bits and one parity bit results.

The 620 inserts correct vertical parity in all transmitted characters and checks vertical parity in all received characters. Receipt in a message sequence from the CPU of a character whose vertical parity is incorrect results in display of an error character on the screen.

In the response mode, detection of a vertical parity error in any character from the CPU results in transmission of a NAK to the CPU after completion of the transmission from the CPU.

The longitudinal redundancy character used in the response modes is the half-add of each of the data bits and the parity bit of all characters (except the STX) in the message sequence. Since the parity bit of the LRC is the half-add of the parity bits of the previous characters, the LRC does not necessarily contain correct vertical parity. Detection of a longitudinal redundancy error by the 620 results in the transmission of a NAK to the CPU. Also the CPU should send a NAK to the 620 if such an error is detected in a 620 message sequence transmission.

EDIT OPERATION

GENERAL

The 620 system has three modes of data entry, defined as 1. Format Type, 2. Type, and 3. Conversation Type. The standard 620 is capable of Format Type mode only. Type mode and Conversation Type mode are available as options.

FORMAT TYPE MODE (FT)

The system is placed in Format Type mode upon initial system turn on and remains in this mode at all times until either a Type mode or Conversation Type mode character is entered from either the keyboard or CPU. The 620 can be returned to Format Type mode by entering a Format Type character.

(FT) Clear (CL) - Upon receipt of this code the contents of memory are cleared and the Memory Start (MS) character is inserted in the memory by the hardware. The cursor is positioned at MS+1.

(FT) Alphanumeric Characters - Each character is entered into memory at the present cursor location. Cursor is front spaced one position except in the case where the character is entered into the location immediately preceding memory start (MS-1). In this case, the cursor remains at this location. All alphanumerics in memory are displayed except an alphanumeric inserted into the 65th position on a line.

(FT) Carriage Return (CR) - Same as alphanumeric, except that the CR is not displayed. This character moves cursor down one line and returns cursor to left margin. If a CR is not inserted in memory an automatic carriage return is accomplished on the display after 64 characters have been inserted on one line and one memory location corresponding to the 65th character position on a line is lost.

(FT) Horizontal Tab (HT) - This character is entered into memory at the present cursor location. Cursor is front spaced one position in memory except in the case where the character is entered into the location immediately preceding memory start (MS-1). In this case, the cursor remains at MS-1. The HT character will cause a four-character space to occur on the display. The HT character is optional and if sent to the 620 without the option, it will be disregarded. If an HT character is entered into the 61st through 64th character position of a line, an automatic carriage return will occur.

(FT) Vertical Tab (VT) - Same as HT except that the VT will cause the display to increment four lines and return to the left margin.

(FT) Start Delta (▷) and Stop Delta (◁) - These characters are entered into memory at the present cursor location. Cursor is front spaced one position except in case where the character is entered into location MS-1. In this case, the cursor remains at MS-1. The start and stop characters are only available with the Format Option and if sent to the 620 without the option, will be disregarded.

(FT) End of Text (ETX) - This character is entered into memory at the present cursor location. Cursor is front spaced one position except where ETX is entered into location MS-1. In this case cursor remains at MS-1.

Cursor Moves - The following characters accomplish cursor movements only and are not entered into memory.

(FT) Reset Cursor (RC) - Cursor is moved from present position to MS+1.

(FT) Front Space (FS) - The Cursor is moved forward one position except in case where present position is at MS-1. In this case no cursor action is accomplished.

(FT) Back Space (BS) - The cursor is moved backward one position except in case where present position is at MS+1. In this case, no cursor action is accomplished. The back space character is not normally used in CPU messages.

(FT) Send (SP) - This character is not inserted into memory. Upon receipt of this code, the 620 enters an output routine. The cursor is moved from its present position to MS+1 and the 620 awaits a CPU poll. Upon receipt of a poll, all characters in memory, up to ETX are sent and cursor is left on ETX character at end of operation. If no ETX is found, all characters in memory, including nulls, are sent and the cursor is left at location MS-1. If the 620 contention option is selected, the poll from the CPU is not required.

TYPE MODE (TP)

Type mode is available as an option in the 620 system, (Format Option). The Format Option enables the 620 to establish a two-block configuration, one fixed and one variable. This option requires the use of Format Type mode and Type mode operations. The fixed block is established in Format Type mode and the variable data is entered in Type mode. A variable field is defined as that data between a start delta character and a stop delta character. Any number of variable fields may be established in Format Type mode by the

use of the start and stop delta characters. The 620 is placed in Type mode by the receipt of a Type character and remains in Type mode until receipt of a Format Type or Conversation Type character. Upon receipt of a Type mode character the cursor is positioned in the first character position of the first variable field in memory, (i.e., $\triangleright + 1$). If there are no variable fields in memory the cursor will stop on MS-1.

Selective Erase

(TP) Clear (CL) - Upon receipt of this code all data in variable fields is deleted from memory. The delta characters are not deleted. The cursor is left at the first $\triangleright + 1$ position upon completing the operation. If there are no variable fields in memory, the cursor is left at MS-1.

(TP) Alphanumeric Characters - Character is entered into memory at the present cursor location. The cursor is front spaced one character if the character is entered into a location not immediately preceding a stop delta or MS. If the character is entered into a location immediately preceding a stop delta, the cursor is front spaced to the first alphanumeric character position in the next variable field in memory. If the character is entered at MS-1, the cursor remains at this position.

(TP) Carriage Return (CR) - No operation

(TP) Horizontal Tab (HT) - No operation

(TP) Vertical Tab (VT) - No operation

(TP) Start Delta (\triangleright) and Stop Delta (\triangleleft) - These characters are used in Type mode to accomplish automatic cursor movement and are not entered into memory. The stop delta is used, after resetting the cursor, to address a particular variable field. The cursor will advance one variable field for each stop delta programmed. The start delta is used by a keyboard operator in Type mode to back the cursor through each variable field. The cursor will back up one variable field for each start delta programmed. The start delta is not normally used in a CPU program as extensive timing restrictions result. The stop delta may also be used in Type mode to reduce transmission requirements. For example, if a formatted block consists of 30 characters and only 10 characters are actually used in the variable field, a stop delta may be used to position the cursor to the beginning of the next variable field, instead of programming 20 front spaces or 20 space codes.

(TP) Start Delta (\triangleright) - Upon receipt of this code the cursor is moved from its present location to the last alphanumeric character location in the preceding variable field in memory. The cursor will search past MS looking for the preceding stop delta in memory. If there are

no stop delta characters in memory, the cursor will stop at location MS+1, the second time through memory.

(TP) Stop Delta (<1) - Upon receipt of this code the cursor is moved from its present location to the first alphanumeric character location in the following variable field in memory. The cursor will search past MS looking for the following start delta character in memory. If there are no start delta characters in memory, the cursor will stop at location MS-1, the second time through memory.

(TP) End of Text (ETX) - Same as Alphanumeric characters in Type mode.

*Not entered
from CPU*

Cursor Moves - The following characters are not entered into memory but are used to accomplish cursor movements. The back space is not normally used in a CPU program as extensive timing restrictions result.

(TP) Reset Cursor (RC) - Upon receipt of this code, the cursor is moved from its present location to the first alphanumeric character location in the first variable field in memory. If there are no variable fields, the cursor is positioned at MS-1.

(TP) Front Space (FS) - If the cursor is positioned on the last character location in a variable field, it is automatically front spaced to the first character location in the next variable field. If the cursor is positioned at MS-1, it remains there. If the cursor is positioned neither at the last character location in a variable field nor at MS-1, it is front spaced one position.

(TP) Back Space (BS) - If the cursor is positioned on the first character location in a variable field it is back spaced to the last character location in the preceding variable field. If the cursor is positioned at MS-1, it remains there. If the cursor is positioned neither on the first character location in a variable field nor at MS-1, it is back spaced one position.

(TP) Send (SP) - This character is not inserted into memory. Upon receipt of this code the 620 enters an output routine. The cursor is moved from its present location to the first alphanumeric position of the first variable field in memory and awaits a CPU poll. Upon receipt of poll, all alphanumeric data located in all variable fields including all stop deltas are sent to the CPU up to ETX and cursor is left on the ETX character. During the output routine, the cursor is automatically advanced from the end of one variable field to the beginning of the next variable field. During the automatic cursor advance between variable fields, null characters may be sent to the CPU dependent on the number of character spaces

between variable fields. If nulls are required, they will be automatically generated by the 620. If no ETX character is encountered, the cursor is left at location MS-1. If the 620 contention option is selected, no poll is required from the CPU.

CONVERSATION TYPE MODE (CT)

Conversation Type mode is available as an option in the 620 system. This option enables the 620 to transmit and receive sequential data down the face of the CRT without disturbing previously transmitted or received data.

Conversation Type mode is accomplished by tagging each character as it is entered into memory. Both CPU data and keyboard data are tagged. Each message segment is terminated with an ETX.

When accomplishing a send routine, all tagged characters up to ETX are sent to the CPU. Upon receipt of an ACK from the CPU, all tag bits in memory are cleared, the ETX in memory is changed to a CR and the cursor is positioned on the CR+1 memory location. If a NAK is received from the CPU, the 620 will not clear the tag bits nor change the ETX to a CR, but will re-transmit the tagged characters and will repeat this sequence for each NAK received until an ACK is received. When accomplishing a write routine to the 620, each CPU character is tagged when entering memory. If the 620 received the message successfully (i.e., no LRC or parity errors) it will send an ACK to the CPU, clear all tag bits and change the ETX character terminating the CPU message to a CR and front space the cursor one position. If an error is detected in the received message, the 620 will send a NAK to the CPU, position the cursor on the first tagged character location, and clear all tagged characters in memory, in anticipation of re-transmission of the message by the CPU.

The 620 is placed in Conversation Type mode by the receipt of a Conversation Type character and remains in Conversation Type mode until receipt of a Format Type or Type character.

(CT) Clear (CL) - Identical operation as in Format Type mode.

(CT) Alphanumeric Characters - Same as Format Type operation.

(CT) Carriage Return (CR) - Same as Format Type operation.

(CT) Horizontal Tab (HT) - Same as Format Type operation.

(CT) Vertical Tab (VT) - Same as Format Type operation.

(CT) Start (▷) and Stop (◁) Delta - Same as Format Type operation.

(CT) End of Text (ETX) - Same as Format Type operation.

Cursor Moves - The following characters accomplish cursor movements only and are not entered into memory.

(CT) Reset Cursor (RC) - The cursor is moved from its present position to the first tagged character in memory. If there are no tagged characters in memory, the cursor is positioned at MS-1.

(CT) Front Space (FS) - Same as Format Type operation.

(CT) Back Space (BS) - Same as Format Type operation.

(CT) Send (SP) - This character is not entered into memory. Upon receipt of this code, the 620 enters an output routine. The cursor is moved to the first tagged character in memory and awaits a CPU poll. Upon receipt of a poll all tagged characters up to ETX, or if no ETX, all tagged characters up to MS are sent to the CPU. Other operations are accomplished dependent on an ACK or NAK reply from the CPU. (See explanation under Conversation Type mode above.) If 620 contention option is selected, no poll is required from the CPU.

DESCRIPTION OF HARD-COPY OPTION

The function of this option, named the Hard-Copy I Option, is to produce a permanent record of data displayed on the Model 620 terminal by copying the data on a Model 33 or Model 35 Teletype.

HARD-COPY OPTION COMPONENTS

The Hard-Copy I Option consists of the following components:

- a. Hard-Copy Buffer I
- b. Printer
- c. Hard-Copy Data Transmission Cable
- d. Modified Keyboard

Hard-Copy Buffer I

The function of the Hard-Copy Buffer I is to accept data from the 620 memory and to control the transfer of the data to the Hard-Copy Printer. The data is transferred serially at the rate of 110 bits per second (10 characters per second). Each character transferred is composed of eleven bits: 1 start bit, 7 USASCII data bits, 1 unused parity bit, and 2 stop bits.

Printer

The printer used with the Hard-Copy I Option is either an ASA 33FR or an ASA 35SP. It is a receive only device which accepts data at a rate of 110 bits per second.

Transmission Cable

A two-wire, twisted-pair cable is provided with the Hard Copy I Option to transfer the data from the 620 to the teletype. The maximum length of this cable is 1000 feet.

Keyboard

The standard 620 keyboard is modified with the Hard-Copy I Option to include a Copy key whose function is to enable an operator to initiate a hard-copy routine. A hard-copy busy lamp is provided that illuminates upon depression of the Copy key and extinguishes upon completion of the copy routine.

HARD-COPY FUNCTIONAL OPERATION

A copy routine may be initiated from either the CPU or the 620 keyboard. In order for the CPU to initiate a hard-copy routine the Copy code (0000011 in bits 1 through 7 respectively) must be contained in the body of the message, (i. e., anywhere between STX and ETX), of a normal CPU write routine. When the message has been positively acknowledged by the 620, the copy routine will be initiated. In order for an operator to initiate a hard-copy routine he must depress the Copy key on the 620 keyboard.

The hard-copy routine is under control of the 620 Edit Logic and therefore the display may not be used while the copy routine is in process. Both the 620 I/O section and the keyboard are interlocked for the duration of the copy routine. During a copy routine, the 620 I/O section will respond to all write and poll requests with an EOT.

A copy routine will terminate upon detection of either an end-of-text (ETX) or a Memory Start (MS) character in the 620 memory, whichever occurs first.

Hard-Copy Code Conversions

The following code conversions are accomplished by the 620 hardware during a hard-copy routine:

- a. Start Delta (▷) is converted to a space code.
- b. Stop Delta (◁) is converted to a space code.
- c. Horizontal Tab (HT) is converted to four space codes.

- d. Vertical Tab (VT) is converted to one carriage return (CR) and four line-feed codes.
- e. Carriage Return (CR) is converted to a carriage return plus a line-feed code.

NOTE

A carriage return and line-feed code is automatically inserted by the 620 hardware at the beginning of each hard-copy routine.

Hard-Copy Modes of Operation

There are two modes of operation for the 620 Hard-Copy routine. Copy Mode 1 copies all data on the 620 CRT and Copy Mode 2 copies only the variable data (i.e., the data between Start and Stop Deltas). The selection of the appropriate copy mode is determined by the 620 mode of operation. When the 620 is in Format Type (FT) or Conversation Type (CT) mode, Copy Mode 1 is selected by the hardware. When the 620 is in Type (TP) mode, the hardware selects Copy Mode 2.

Mode 1 Operations: Copy Mode 1 copies all data sequentially received from the 620 with the code conversions as outlined in the paragraph above entitled "Hard-Copy Code Conversions".

Mode 2 Operation: Copy Mode 2 copies all data sequentially received from the 620 with the code conversions as stated in the paragraph above entitled "Hard-Copy Code Conversions" as well as with the following conversions:

- a. All alphanumeric data (i.e., this does not apply to CR, VT, and HT) received within the following boundaries are converted to space codes:
 - 1. All A/N data between Memory Start (MS) and the first Start Delta (▷).
 - 2. All A/N data between a Stop Delta (◁) and subsequent Start Delta (▷).
 - 3. All A/N data between the last Stop Delta (◁) and Memory Start (MS).
- b. All null codes received between a Start Delta (▷) and the following Stop Delta (◁) are converted to space codes.

SECTION 4
PROGRAMMING CONSIDERATIONS FOR USE OF 620
WITH IBM SYSTEM/360

REMOTE TRANSMISSION - (2400 bits per second or less)

In general, IBM provides a read/write routine common to all supported TP devices plus a number of Device Dependent Modules (DDM's), each of which consists of a tabular set of constants uniquely defined for a supported device. At OPEN time, a logical connection is established between the appropriate DDM, the I/O control block (DCB, DTFQT, or DTFBT), and the common read/write routine. Subsequent I/O requests, via the language (i. e., macro instruction) facilities of the IBM access method used, cause the common read/write routine to generate channel programs appropriate to the device to be communicated with by using the reference information tabulated in the linked DDM.

The 620 Display Terminal becomes fully supported under any IBM access method in this category by simply using the appropriate Sanders - furnished Device Dependent Module(s). The Sanders - furnished DDM(s) may be incorporated and put to use in a number of ways. For example, in the event that the only display terminal type to be used in a particular installation is the Sanders 620 Data Display System, it is only necessary to replace the DDM intended for the IBM 2260 with the Sanders - furnished DDM for the 620. The full facilities of the access method now become available and can be used exactly as described in the appropriate IBM publication.

OS BTAM - (REMOTE CONNECTION)

The following discussion describes the steps one must follow to use the 620 Data Display System with OS BTAM.

The problem programmer must provide a Data Control Block (DCB) for each line group to be used. He must initialize each DCB by using the IBM OPEN macro-instruction and should insure that each OPEN operation was successful. After execution of the OPEN macro-instruction, bit 3 of the DCBOFLGS field in the DCB, dcbaddr + 48, is set to 1 if the Data Control Block has been opened successfully and is set to 0 if the DCB has not been

opened successfully. If the user has not replaced the catalogued IBM 2260 DDM (IGG019M3) in SYS1.SVCLIB, he then must issue the DOPEN macro-instruction (provided by Sanders) to set up the necessary linkages between OS BTAM, one of the Sander's DDM's, and the DCB's in his program. (See Sanders First-Level Support Manual for a detailed description of the DOPEN macro-instruction.) If replacement of the IBM 2260 DDM is chosen, the IBM utility ILEHIOSUP must be run to update the TTR's in the SVCLIB directory.

In addition to using the DOPEN macro-instruction, the problem programmer should use the EBCDIC to ASCII-8 and ASCII-8 to EBCDIC translate tables provided by the Sanders' TABLES macro-instruction. (See Sanders First-Level Support Manual for a detailed description of the TABLES macro-instruction.) In addition to the normal alphanumerics, these tables contain the special edit and function characters unique to Sanders Displays. (See Appendix F of this publication.)

OS QTAM - (REMOTE CONNECTION)

To use the 620 Display System with OS QTAM, the user should do the following:

- a. Use the Sanders-supplied tables, provided by the TABLES macro-instruction, for translating incoming and outgoing messages. (See Sanders First-Level Support Manual for a detailed description of the TABLES macro-instruction.)
- b. Ensure that when the BTAM read/write routine is invoked by QTAM for an I/O operation, the routine will construct channel programs compatible with the Sanders 620 Display System.

To accomplish step "a" above, two macro-instructions, one named TRANS provided by IBM and another named TABLES provided by Sanders, may be used. (See IBM publication C30-2003 for an explanation of TRANS.)

There are two methods available for accomplishing step "b":

Method 1:

If the user is operating in a non-mixed environment, i.e., is using one model-type of Sanders Display, he may replace the IBM 2260 DDM with the appropriate Sanders-supplied DDM by link-editing the Sanders DDM as a replacement for the module IGG019NR in SYS1.SVCLIB and running the IBM utility program IEHIOSUP to update the TTR's in the SVCLIB directory.

Method 2:

If on the other hand, the user wishes to retain the capability for operating IBM 2260's or he wishes to operate more than one model type of Sanders display, he must first link-edit the Sanders DDM into SYS1.LINKLIB. Secondly, he must employ the DOPEN macro-instruction to perform the necessary DCB initialization. Finally, he must furnish the linkage editor control statements to include the modules in his program.

NAME	OPERATION	OPERAND	72
OPENIO	OPEN	(DISK,(INOUT), LINE2260, (INOUT), LINE620, (INOUT, IDLE))	X X X
	DOPEN	LINE620, DEVTYPE=(620, QTAM)	X X
	LA	7, LINE620	
	STARTLN	(7)	

Note that using the IDLE parameter in the "options" field associated with the Sanders display line group defers polling of the line group to allow DCB initialization to be performed by DOPEN. Even if IGG019NR is replaced, the IDLE option should be used.

Subsequently, polling is initiated by STARTLN. (See Sanders First-Level Support Manual for a detailed description of the DOPEN macro-instruction.)

To obtain automatic linking of the DOPENed DDM's, ensure that the NCAL option is not specified and that the JCL for the linkage editor includes the following DD card:

NAME	OPERATION	OPERAND	
//SYSLIB	DD	DSNAME = name*, DISP = OLD	X

* name = the name of the data set containing the DDM.

DOS BTAM - (REMOTE CONNECTION)

To employ the 620 Display System in conjunction with DOS BTAM, the user must do the following:

a. Use the translate tables provided by Sanders to convert incoming and outgoing messages to the appropriate codes.

b. Ensure that the DOS BTAM control program will access the appropriate Sanders-supplied Device Dependent Module (DDM) for constructing channel programs.

c. Ensure that all DTFBT's which refer to Sanders display line groups have been properly initialized.

To accomplish the first requirement, a translate table generating marco-instruction named TABLES has been supplied. (See Sanders First-Level Support Manual for a detailed description of the TABLES macro-instruction.)

The second and third requirements as stated above may be met by using the DOPEN macro-instruction. (See the description of the DOPEN macro-instruction in Sanders First-Level Support Manual.)

DOS QTAM - (REMOTE CONNECTION)

To use Sanders 620 Display System with DOS QTAM, the user should do the following:

a. Use the translate tables provided by Sanders to convert incoming and outgoing messages to the appropriate codes.

b. Ensure that the DOS QTAM control program will access the appropriate Sanders-supplied Device Dependent Module (DDM) for constructing channel programs. (Refer to the description of the DOPEN macro-instruction in Sanders First-Level Support Manual.)

To accomplish the first requirement, a translate table generating macro-instruction, TABLES, is supplied by Sanders. (See Sanders First-Level Support Manual for a detailed description of TABLES.)

There are two methods available for accomplishing step "b":

Method 1:

If the user is operating in a non-mixed environment, i.e., is using one model-type of Sanders Display, he may replace the IBM 2260 DDM for QTAM with the appropriate Sanders-supplied DDM in the relocatable library.

Method 2

If the user wishes to retain capability for operating IBM 2260's or he wishes to operate more than one type of Sanders display, he must employ the DOPEN macro with DEVTYPE = ({620 or 720}, Q [TAM]) to perform the necessary DTFQT initialization.

DEVICE DEPENDENT MODULE (DDM) DESCRIPTION

A number of Device Dependent Modules have been designed by Sanders to provide I/O processing compatibility between OS BTAM/QTAM, DOS BTAM/QTAM, and Sanders Display Systems in a remote environment.

DETAILED MESSAGE STRUCTURES - Remote Operation

Messages received from a 620 terminal by the CPU have the following format:

STX	X '02'	Start of Text
CUA		Control Unit Address (1 byte)
X '09'	Trans. to X '4A'	620 Display Address
First Text Character	}	Variable length message from 620 memory
.		
.		
.		
Last Text Character		
ETX	X '03'	End of Text
LRC		Longitudinal Redundancy Character
EOT	X '04'	

NOTE

The LRC character is sent by the 620 system but is not transferred into main storage.

Messages sent to a 620 terminal from the CPU have the following format:

	STX	Start of Text	
	FT	Format type or other mode-change character	
I/O Area	First Text Character	}	Variable length message from user's I/O area
	.		
	.		
	.		
	Last Text Character		
	ETX		

ETX

End of Text

LRC

Longitudinal Redundancy Character

NOTE

The LRC character is generated by the transmission control unit - no user action is required.

ACCESS-METHOD DEPENDENT INFORMATION

Access Method	Messages <u>from</u> 620 to CPU	Messages <u>to</u> 620 from CPU
OS BTAM	STX appears in first-character position of user's area followed by CUA, X '09' (display address), and message text; terminated by an ETX.	STX and FT on output are provided by the DDM. The user can start his message in position one of his buffer. If a mode other than FT is desired, the user can start with another mode-change character.
OS QTAM	Same as above.	Same as above.
DOS BTAM	Same as OS BTAM.	Under DOS BTAM the user <u>must</u> provide his own STX character and FT or other mode-change character. The message must terminate with an ETX.
DOS QTAM	Same as OS QTAM.	Same as OS QTAM.

USING A DDM

A user may obtain the services of a particular DDM in one of two ways:

- a. He may replace the appropriate IBM 2260 DDM with the desired Sanders DDM, renaming it with the name of the 2260 DDM.
- b. Use the DOPEN macro-instruction (See Sanders First-Level Support Manual for a detailed explanation of DOPEN).

The DEVTYPE parameter of the DOPEN macro-instruction should be coded as follows to obtain the services of a particular DDM:

* Translated to X'4A' in case of on disk

DEVTYPE = (model-number, access-method)

Model-Number	Access-Method
720 or 620	BTAM
720 or 620	QTAM

OS-DDM'S

(BTAM) DDMOSN01
(QTAM) DDMOSQ01

DOS-DDM'S

DDDOSN01
DDDOSQ01

MESSAGE PROCESSING MACROS

A number of message processing macro-instructions have been designed by Sanders to assist the user of Sanders Data Display Systems in handling his message processing functions. Of particular interest to the 620 user are two macros named EDITMOVE and TABLES.

EDITMOVE MACRO-INSTRUCTION

The facilities provided by EDITMOVE are embodied in the EDITMOVE macro-instruction and the EDITMOVE processor. Both the macro-instruction and the processor are equally compatible to OS and DOS users. The EDITMOVE processor will move a stream of input characters to an output area and perform specified serial editing operations in transit. The range of editing operations performed with a single pass through the input stream includes substitutions, insertions, and deletions caused by user specified characters and/or character-string sequences encountered in the input stream. If the editing operations to be performed are known in advance (i.e., at compile time), the EDITMOVE macro-instruction can be used to specify the desired editing operations in a systematic way with standard notation conventions. If editing specifications must be derived at execution time, parameters may be dynamically constructed for the EDITMOVE processor without using the EDITMOVE macro-instruction. (See Sanders First-Level Support Manual for a full explanation of EDITMOVE parameters and specifications.)

TABLES MACRO-INSTRUCTION

Use of the TABLES macro-instruction causes generation of an EBCDIC to ASCII-8 translate table and/or the generation of an ASCII-8 to EBCDIC translate table. (See Translation Tables in Appendix F of this publication.) These tables should be used to translate messages sent to and received from a Sanders 620 Data Display System. (See Sanders First-Level Support Manual for a full explanation of the parameters for the TABLES macro.)

SECTION 5
TIMING CONSIDERATIONS FOR 620 CONTROL OPERATIONS

Timing considerations within the 620 System are mode dependent and thus will be discussed on an operating mode basis. The memory access time of the 620 is 5.55 milliseconds and the majority of functions are able to be completed in one memory cycle time. Some control functions will require additional time to complete their operation in certain modes and it will therefore be necessary to program null characters. The number of null characters required is a function of the mode (asynchronous or synchronous) and transmission rate (110, 1000, 1200, 1800, 2000, 2400, etc.) of data transmission. The transmission time taken by the number of null characters used must equal the fixed response time required to complete the desired 620 control operation. To facilitate the editing operations required for null character insertion in outgoing messages, use of the Sanders-furnished EDITMOVE macro-instruction (described elsewhere in this publication) is recommended.

The formulas below will enable the programmer to determine the worst case number of nulls to insert in a message for each control character. In some instances it will be noted that the number of nulls needed must be partially estimated due to unpredictability of the contents of the 620's memory.

NOTE

The notation (X)Y means "Y, in mode X",
where X is a mode change character (i. e.,
FT, TP, or CT).

How to Use the Formulas: Unless otherwise stated the quantity of nulls represented should be inserted following the given character in the message.

- a. Determine the value of f from the table applicable to the current transmission mode and transmission rate.
- b. f represents a number dependent on transmission characteristics which is then used in conjunction with the formulas to determine the number of nulls to be inserted. Where multiples of f are indicated, round down the final sum or product only. Otherwise, f may

always be rounded down to a whole number. (e.g., if the final sum or product of a calculation equals 6.86 nulls, this may be rounded down to 6 nulls.)

c. To determine the fixed response time required for any of the listed 620 control operations which require nulls, set f equal to 5.55 milliseconds. Thus (FT) BS = f = 5.55 milliseconds.

CONTROL OPERATIONS TIMING

FORMAT TYPE MODE

The following timing constraints are placed on programming the 620 in Format Type (FT) mode:

- (FT)CL = no null requirement
- (FT)A/N = no null requirement
- (FT)CR = no null requirement
- (FT)HT = no null requirement
- (FT)VT = no null requirement
- (FT)▷ and ◁ = no null requirement
- (FT)ETX = no null requirement
- (FT)RC = no null requirement
- (FT)FS = no null requirement
- (FT)BS = f after each BS in FT mode
- (FT)SP = no null requirement

TYPE MODE

The following timing constraints are placed on 620 programming when in Type (TP) mode:

(TP)CL = $5f$ after each CL in TP mode

(TP)A/N = no null requirement except when the A/N entry is made in a location immediately preceding a Stop Delta (◁) character. In this case, the cursor is automatically advanced to the beginning of the next variable field, and the null requirement equals $f/260X$ (number of characters through which cursor must advance). *3f for full screen (780 fsc)*

(TP)CR = no operation in TP mode

(TP)HT = no operation in TP mode

(TP)VT = no operation in TP mode

(TP)▷ and ◁ : The timing requirements with respect to Start and Stop Delta are functions of the format already established in memory.

(TP)▷ = f for each character through which the cursor must backspace. The use of this code requires that a variable number of nulls be programmed depending upon the number of positions in memory through which the cursor is automatically backspaced. The exact number of backspaces is a function of the present cursor location and the location of the preceding Stop Delta (◁) character in memory.

(TP)◁ = $f/260 \times$ (number of characters through which cursor must advance). The use of this code causes an automatic forward space of the cursor which can be accomplished in 21.3 microseconds per character position. The exact number of forward spaces is a function of the present cursor location and the location of the following Start Delta (▷) character in memory.

(TP)ETX = no null requirement

(TP)RC = $f/260 \times$ (number of characters through which cursor must advance); the total number of characters is measured from MS (Memory Start) to the first Start Delta (▷) character in memory.

(TP)FS = no null requirement except when the FS entry is made in a location immediately preceding a Stop Delta (◁) character. In this case, the cursor is automatically advanced to the beginning of the next variable field, and the null requirement equals $f/260 \times$ (number of characters through which cursor must advance).

(TP)BS = f after each BS in TP mode except when the cursor is located immediately following a Start Delta (▷) character. In this case, the null requirement equals f for each character through which the cursor must backspace to get to the last character of the preceding variable field, as in the operation of (TP)▷ above.

(TP)SP = no null requirement

CONVERSATION TYPE MODE

The following timing constraints are placed on 620 programming when in Conversation Type (CT) mode:

(CT)CL = no null requirement

(CT)A/N = no null requirement

(CT)CR = no null requirement

(CT)HT = no null requirement

(CT)VT = no null requirement

(CT)▷ and ◁ = no null requirement

(CT)ETX = no null requirement

(CT)RC = $f/260 \times$ (number of characters through which cursor must advance);
the total number of characters is measured from MS (Memory Start) to the first character
in the current message.

(CT)FS = no null requirement

(CT)BS = f after each BS in CT mode

(CT)SP = no null requirement

TABLE OF f VALUES

Transmission Rate	Transmission Mode		
	Synchronous 8-bit	Asynchronous	
		10-bit	11-bit
2400	1.67	1.33	1.21
2000	1.39	1.11	1.01
1800	N/A	1.00	0.908
1200	N/A	0.666	0.605
1000	N/A	0.555	0.505
110	N/A	0.061	0.056

SECTION 6 INTERFACE REQUIREMENTS AND CHARACTERISTICS

HARDWARE INTERFACE

The 620 Data Display System is compatible with and meets the requirements of EIA Standard, RS-232-B. The 620 contains selectable interfaces consisting of jumper wire patching on the Modem Control Card enabling it to interface with either a data set or the Sanders 716 Serial Distributor.

DATA SET INTERFACE

Examples of typical data sets that the 620 is designed to interface with are the following AT & T data sets:

- a. Model 103A
- b. Model 103F
- c. Model 202C
- d. Model 202D
- e. Model 201A
- f. Model 201B

SERIAL DISTRIBUTOR INTERFACE

The 620 may also be interfaced with the Sanders 716 Serial Distributor. The 716 Serial Distributor enables the 620 to be located up to 1000 feet away from the data set and allows multiple, (up to sixteen), 620's to be connected to a single data set. The primary difference in the interface is that the signals are +5 V and 0 V instead of the data set levels of ± 15 V. When the Serial Distributor is used, the Data Terminal Ready Signal originates in the Serial Distributor and is not individually carried from each 620 to the Serial Distributor.

SOFTWARE INTERFACE WITH IBM SYSTEM/360

The following are special system generation considerations for interface of the Sanders 620 Data Display System with IBM System/360:

SPECIAL OS SYSTEM GENERATION CONSIDERATIONS

BTAM and QTAM - Remote Connection

The following system generation macros should be completed for utilization of the 620 with BTAM and QTAM:

	NAME	OPERATION	OPERAND
As Required For 731 *For 620 Line Groups		DATAMGT	
		TELCMLIB	
		IOCONTRL	UNIT = 2701, ADDRESS = XX
		IODEVICE	UNIT = 2260, ADAPTER = IBM 3, X MODEL = 1, ADDRESS = XXY
		IODEVICE	UNIT = 2260, ADAPTER = IBM 3, X MODEL = 1, ADDRESS = XXY

*(Up to eight 620 line groups per 731 Data Communications Buffer)

XX = channel number and channel position (731)

Y = 731 - line position (for 620 display terminal or 716 serial distributor)

Caution

In choosing line addresses for graphic devices, care must be taken to select an address which is acceptable to the channel. An IBM representative should be consulted to determine which addresses have been implemented on the channel you wish to use for the I/O connection.

SPECIAL DOS SYSTEM GENERATION CONSIDERATIONS

BTAM and QTAM - Remote Connection

The DVCGEN system generation macro should be completed as follows for utilization of the 620 with BTAM and QTAM:

NAME	OPERATION	OPERAND
	DVCGEN	CHUN = X' cul' , DVCTYP = 2701

where

c = 0 for Multiplexor Channel

u = 731 Data Communications Buffer number (O-F)

l = 731 line number (for 620 display terminal or 716 serial distributor)

Caution

In choosing line addresses for graphic devices, care must be taken to select an address which is acceptable to the channel. An IBM representative should be consulted to determine which addresses have been implemented on the channel you wish to use for the I/O connection.

SECTION 7
INSTALLATION PLANNING DATA

INPUT POWER REQUIREMENTS

The 620 operates entirely from the following source:

- 117 VAC \pm 10 percent
- 48 to 62 Cycles per second
- 200 Watts

PHYSICAL SPECIFICATIONS

DISPLAY TERMINAL

The model 620 Data Display Terminal is a self-contained desk-top unit. The display cabinet is mounted on a 4-1/4 inch high base at an angle which allows a normal reading attitude when the display terminal is located on a desk top.

Dimensions and Weight

	<u>CABINET</u>	<u>BASE</u>
Height	14 inches	4-1/4 inches
Width	16-1/2 inches	16-1/2 inches
Depth	20-1/2 inches	20-1/2 inches
Weight	50 pounds (approximate)	

Connectors

Two female connectors are mounted on the base of the display terminal. One of these connectors is to connect to either the keyboard or cable from the keyboard. The other is an output connector for connection to a data set.

Cables

A seven-foot three-prong AC cord and a ten-foot data set cable are provided with the 620.

KEYBOARD

The keyboard is in the form of an alphanumeric typewriter keyboard with 64 alpha- numerics and 12 additional function keys. An optional ten-foot cable with connector may be supplied to connect the keyboard to the display terminal. All electronics are contained within the keyboard.

Dimensions and Weight

Height	4-1/4 inches	Depth	6-7/8 inches
Width	16-1/2 inches	Weight	9 pounds (approximate)

Connectors

A single 37-pin male connector on the rear of the keyboard connects either directly to the display terminal or to a cable with a mating connector. This connector carries all input and output signals.

Function Keys

There are 16 function keys, four of which are spare.

EDIT FUNCTIONS	I/O FUNCTIONS	CURSOR CONTROL	OPTIONS
Format Type (FT) Type (TP) Conversation Type (CT) ▷ (Start Delta) ◁ (Stop Delta) Clear (CL)	Send (SP) ETX (End of Text)	Reset Cursor (RC) Cycle Left Cycle Right	Copy (CPY)

ENVIRONMENTAL CONDITIONS

OPERATING

Temperature:	+60 ^o F to +90 ^o F
Humidity:	10 to 90 percent without condensation
Barometric Pressure:	Up to 10,000 feet elevation.

STORAGE

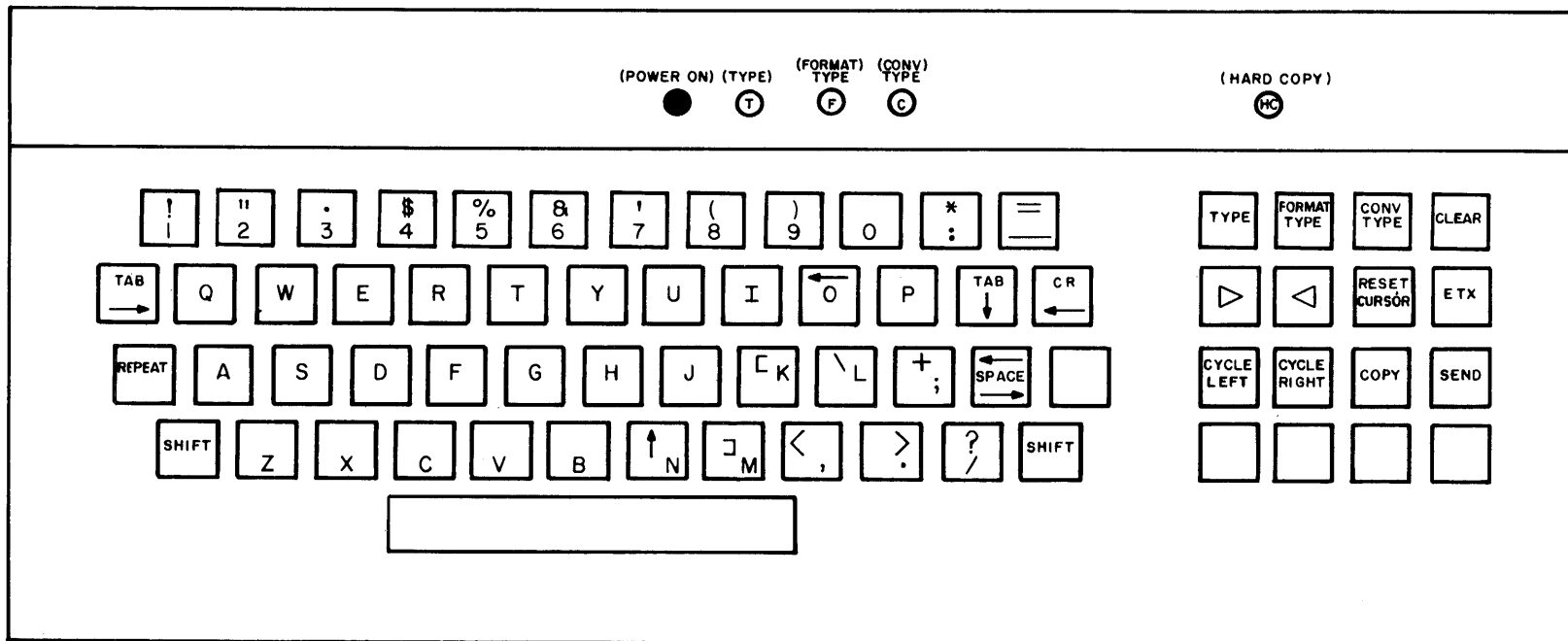
Temperature:	-30 ^o F to +120 ^o F
Humidity:	Up to 90 percent without condensation
Barometric Pressure:	Up to 50,000 feet elevation.

APPENDIX B
620 CODE TABLE

(USASCII SUBSET)

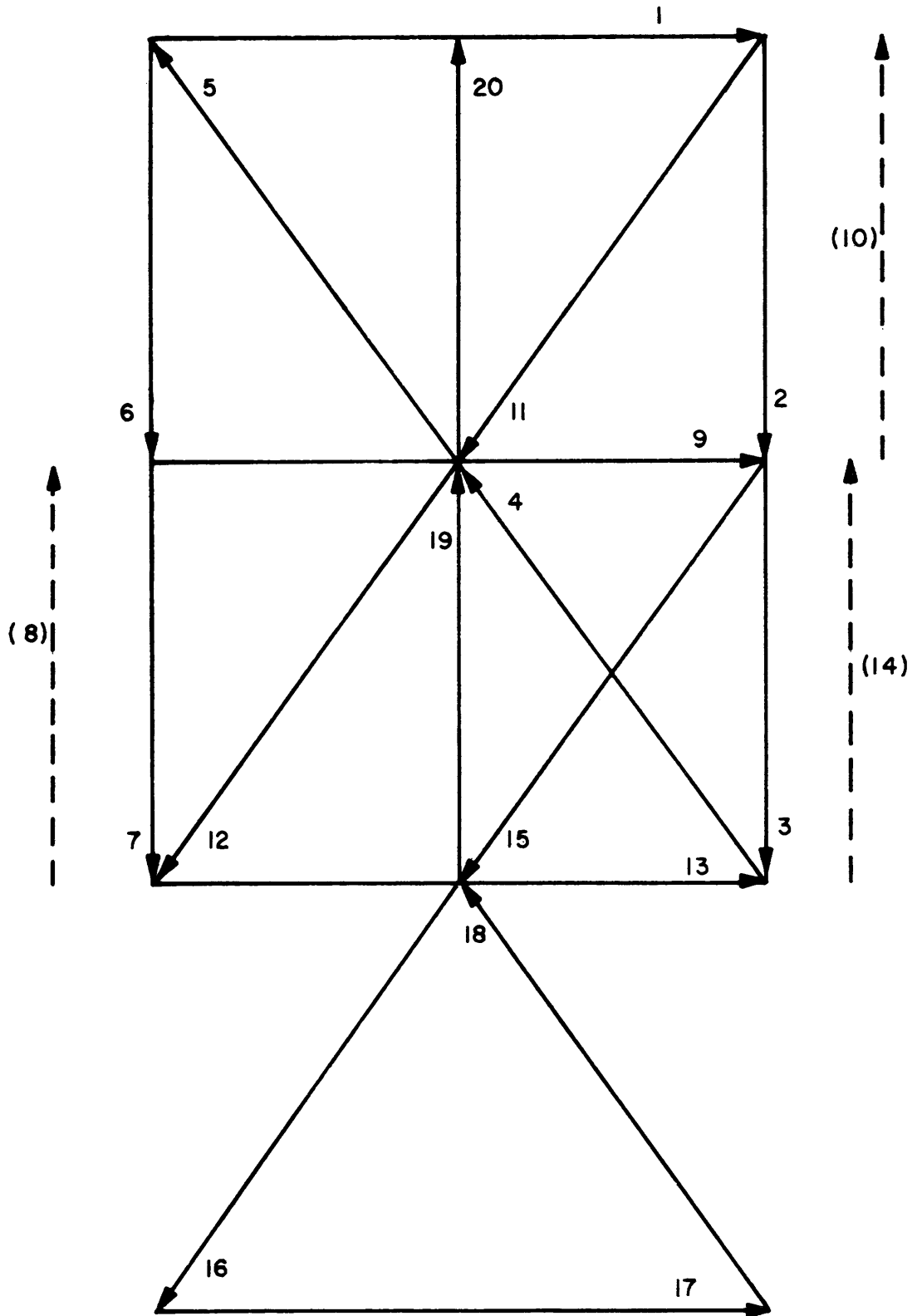
Bit Structure				7	0	0	0	0	1	1	1	1
				6	0	0	1	1	0	0	1	1
4	3	2	5 1	0	1	0	1	0	1	0	1	
0	0	0	0	Null		(Poll) SP	0	(Write) @	P	COPY		
0	0	0	1	SOH		!	1	A	Q			
0	0	1	0	STX		"	2	B	R	TYP		
0	0	1	1	ETX		· (dot)	3	C	S	FORM TYP		
0	1	0	0	EOT		\$	4	D	T			
0	1	0	1		NAK	%	5	E	U			
0	1	1	0	ACK	SYNC	&	6	F	V			
0	1	1	1			' (apos)	7	G	W			
1	0	0	0	FS	BS	(8	H	X			
1	0	0	1	HT)	9	I	Y	SEND		
1	0	1	0			*	:	J	Z	CLEAR		
1	0	1	1	VT		+	;	K	[RES CUR		
1	1	0	0			' (comma)	<	L	\			
1	1	0	1	CR		-	=	M]			
1	1	1	0	▷	FORM FEED	· period	>	N	↑	MS		
1	1	1	1	◁		/	?	O	←	CONV TYP	Parity Error	

APPENDIX C
620 KEYBOARD



APPENDIX D

620 CHARACTER-GENERATOR RACETRACK SCHEMATIC



APPENDIX E

TABLE OF 620 CONTROL OPERATIONS

Character	Format Type Mode	Type Mode	Conversation Mode
Clear (CL)	<ol style="list-style-type: none"> 1. Memory contents cleared 2. Memory start (MS) character inserted in memory 3. $C = MS + 1$ <i>N.S</i> 	<ol style="list-style-type: none"> 1. All data in variable fields is deleted. 2. $C = \text{first start delta} + 1$ 3. If no variable data exists $C = MS - 1$ 	Same as format type
Alpha-numeric (A/N)	<ol style="list-style-type: none"> 1. Character entered at C location in memory \longrightarrow 2. $C = C + 1$ 		
Carriage Return (CR)	<ol style="list-style-type: none"> 1. Character entered at C location in memory 2. C positioned at next line, left margin on screen 	No action	Same as format type
Horizontal Tab (HT)	<ol style="list-style-type: none"> 1. Character entered at C location in memory 2. Causes a four character space on screen 		Same as format type
Vertical Tab (VT)	<ol style="list-style-type: none"> 1. Character entered at C location in memory 2. Causes a four line increment on the screen 3. C returns to left margin 		Same as format type
End of Text (ETX)	<ol style="list-style-type: none"> 1. Character entered at cursor location in memory \longrightarrow 2. Cursor incremented by one character 		
Front Space (FS)	Cursor positioned \rightarrow to next character in memory \longrightarrow		
Back Space (BS)	Cursor positioned \leftarrow to previous character in memory \longrightarrow		
Reset Cursor (RC)	Cursor is moved to (MS + 1) from its present position	Cursor moved from present location to first A/N character in first variable field	Cursor moved from present position to first tagged character in memory
Start and Stop Delta ($\triangleright \triangleleft$)	<ol style="list-style-type: none"> 1. Character entered at C location in memory 2. $C = C + 1$ 	<ol style="list-style-type: none"> 1. Character not entered in memory 2. Used to step through variable data fields. Start to go backward. Stop to go forward. 	Same as format type
C = Cursor MS = Memory Start			

APPENDIX F
COMPARISON TABLE - 620/720 MESSAGE FORMATS

<u>Addressing the 620</u>	<u>Addressing the 720</u>
EOT	EOT
SOH	SOH
CUA	CUA
X'09'	DA
COMMAND (Write or Poll)	COMMAND (Write or Poll)
<u>Writing to the 620</u>	<u>Writing to the 720</u>
STX	STX
CL (optional)	CL (optional)
Message	Message
ETX	ETX
<u>Reading from the 620</u>	<u>Reading from the 720</u>
STX	STX
CUA	CUA
X'09'	DA
Message	HM
ETX	Message
	ETX

APPENDIX G TRANSLATION TABLES

SYM	ASCII-8	SYM	ASCII-8
BB-BEGIN BLINK	‡ 1B	MC-MOVE CURSOR	* E1
BR-BACK RETURN	* 1D	MS-MEMORY START	EE
BS-BACK SPACE	1B	RC-RESET CURSOR	* EB
BT-BACK TAB	* 19	SB-SEND BLOCK	* E8
CL-CLEAR	* EA	SP-SEND PAGE/SEND	* E9
CPY-COPY	* E0	SOH-START OF HEADER	01
CR-CARRIAGE RETURN	00D	▷-START DELTA	0E
CT-CONV. TYPE	* EF	◁-STOP DELTA	0F
DEL-DELETE	* E6	STX-START OF TEXT	02
EB-END BLINK	1C	TYP-TYPE	* E2
EOT-END OF TRANS.	04	VT-VERTICAL TAB	0B
ETX-END OF TEXT	03		
FD-FORMAT DELETE	* E7	RTR-REQ. TO RECEIVE (WRITE)	A0
FI-FORMAT INSERT	* E5	RTS-REQ. TO SEND (POLL)	40
FS-FORWARD SPACE	08		
FT-FORMAT TYPE	* E3	*-MODE CHANGE	
HM-HOME	0C	+ - TYPE MODE ONLY	
HT-HORIZONTAL TAB	09	‡-FORMAT MODE ONLY	
INS-INSERT	* E4		

ASCII-8 TO EBCDIC																	
HEX TENS																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
HEX UNITS	0	NULL			␣	0					@	^	P			CPY	
		00			40	F0					7C	D7					
	1	SOH			!	1					A	Q					MC
		01			5A	F1					C1	D8					
	2	STX			"	2					B	R					TYP
		02			7F	F2					C2	D9					
	3	ETX			(DOT)	3					C	S					FT
		03			70	F3					C3	E2					
	4	EOT			\$	4					D	T					INS
		04			5B	F4					C4	E3					
	5				%	5					E	U					FI
					6C	F5					C5	E4					
	6				&	6					F	V					DEL
					50	F6					C6	E5					
	7				'	7					G	W					FD
					7D	F7					C7	E6					
8	FS	BS		(8					H	X					SB	
	08	18		4D	F8					C8	E7						
9	HT	BT)	9					I	Y					SP	
	4A	19		5D	F9					C9	E8						
A				*	:					J	Z					CL	
				5C	7A					D1	E9						
B	VT	BB		+	;					K	[RC	
	4F	6D		4E	5E					D2	BB						
C	HM	EB		,	<					L	\						
	7B	6A		6B	4C					D3	BC						
D	CR	BR		.	=					M]						
	5F	1D		60	7E					D4	BD						
E	▷			.	>					N	↑					MS	
	62			4B	6E					D6	BE						
F	◁			/	?					O	←					CT	
	63			61	6F					D6	BF						

ASCII-7 TO ASCII-8
7654321 → 76x54321

TO EBCDIC
(17×X) 01234567

ACK=06
NAK=15

*E0-EF NEVER RECEIVED BY THE CPU FROM A 720 OR 620

APPENDIX G
TRANSLATION TABLES (Cont)

L.P. 2/12

SYM	PUNCHED CARD CODE	PUNCH CHAR	EBCDIC	SYM	PUNCHED CARD CODE	PUNCH CHAR	EBCDIC
BB	0-8-5		6D	MC	12-11-0-8-1		B0
BR	11-9-8-5		1D	MS	11-0-6-8-9		EE
BS	11-9-8		18	RC	12-11-0-8-2		BA
BT	11-9-8-1		19	SB	12-11-0-7		B7
CL	12-11-0-9		B9	SP	12-11-0-8		B8
CPY	0-2-8		E0	SOH	12-9-1		01
CR	11-8-7		5F	D	11-0-2-9		62
CT	11-0-7-8-9		EF	◁	11-0-3-9		63
DEL	12-11-0-5		B5	STX	12-9-2		02
EB	12-11		6A	TYP	12-11-0-1		B1
EOT	12-9-4		04	VT	12-8-7		4F
ETX	12-9-3		03	[12-11-0-8-3		BB
FD	12-11-0-6		B6	\	12-11-0-8-4		BC
FI	12-11-0-4		B4]	12-11-0-8-5		BD
FS	12-9-8		08	↑	12-11-0-8-6		BE
FT	12-11-0-2		B2	←	12-11-0-8-7		BF
HM	8-3	#	7B	(DOT)	12-11-0		70
HT	12-8-2	⚡	4A				
INS	12-11-0-3		B3				

*Not Std
EBCDIC
ACTABADIA*

		EBCDIC TO ASCII-8															
		HEX TENS															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
HEX UNITS	0	NULL				␣	&	-	(DOT)				MC			CPY	0
		00				40	46	4D	43				E1			E0	50
	1	SOH						/					TYP	A	J		1
		01						4F					E2	A1	AA		51
	2	STX						D					FT	B	K	S	2
		02						0E					E3	A2	AB	B3	52
	3	ETX						◁					INS	C	L	T	3
		03						0F					E4	A3	AC	B4	53
	4	EOT											FI	D	M	U	4
		04											E5	A4	AD	B5	54
	5	*CR											DEL	E	N	V	5
		0D											E6	A5	AE	B6	55
	6												FD	F	O	W	6
													E7	A6	AF	B7	56
	7												SB	G	P	X	7
													E8	A7	B0	B8	57
8	FS	BS										SP	H	Q	Y	8	
	08	18										E9	A8	B1	B9	58	
9		BT										CL	I	R	Z	9	
		19										EA	A9	B2	BA	59	
A					HT	!	EB	␣				RC					
					09	41	1C	5A				EB					
B					e	\$,	HM									
					4E	44	4C	0C				BB					
C					<	*	%	@									
					5C	4A	45	A0				BC					
D		BR			()	BB	'									
		1D			48	49	1B	47				BD					
E					+	;	>	=								MS	
					4B	5B	5E	5D								↑	
F					VT	CR	?	"								←	
					0B	0D	5F	42				BF				EF	

* 15=NL=CR

APPENDIX H
GLOSSARY OF TERMS

ACK	The USASCII Acknowledge character used to inform the originator of a communication that the message has been successfully received.
ASYNCHRONOUS	Having a variable time interval between successive bits or characters.
BTAM	Basic Telecommunications Access Method for IBM System/360.
CONTENTION	A condition on a multidrop communication channel when two or more locations try to transmit at the same time.
CPS	Characters Per Second.
CRT	Cathode Ray Tube.
DATA SET	A modulation or demodulation device designed to convert electrical pulses or bits to signals that can be carried over a transmission circuit and vice versa.
DOS	Disk Operating System for IBM System/360.
EBCDIC	Extended Binary Coded Decimal Interchange Code for IBM System/360.
EOT	The USASCII End-of-Transmission character used to denote the end of a data transmission to or from a remote terminal.
ETX	The USASCII End-of-Text character used to denote the end of the message portion of a data transmission.
LRC	Longitudinal Redundancy Character accumulated by a longitudinal redundancy check over all characters from, but not including, STX up to and including ETX.
MODEM	Contraction of Modulator/Demodulator. Modulator is a device that converts electrical pulses or bits (DC signals) to tones (AC signals) which may be carried over a transmission circuit. Demodulator is a device that converts tones from a transmission circuit to electrical pulses or bits suitable for a teletype device or other business machine.
NAK	The USASCII Negative Acknowledge character used to inform the originator of a communication that the message has <u>not</u> been successfully received.

NON-RESPONSE MODE	The non-use of message acknowledgements (ACK and NAK) between receiving and transmitting stations.
NULL	The USASCII Null character (X'00') representing the absence of any information.
OS	Operating System for IBM System/360.
PARITY	A binary check digit (one or zero) carried along in a character or group of characters depending on whether the total number of ones, or zeros, in the character or group is odd or even.
POLL	A flexible, systematic method, centrally controlled, for permitting stations on a multipoint circuit to transmit without contending for the line.
QTAM	Queued Telecommunications Access Method for IBM System/360.
RESPONSE MODE	The use of message acknowledgements (ACK and NAK) between receiving and transmitting stations.
STX	The USASCII Start-of-Text character used to denote the beginning of the message portion of a data transmission.
SYNCHRONOUS	Having a constant time interval between successive bits or characters.

SANDERS REGIONAL SALES OFFICES

BOSTON, MASSACHUSETTS

147 Milk Street
(617) 542-5863

NEW YORK, NEW YORK

529 Fifth Avenue
(212) 687-1050
TWX: 710 581-2066

PITTSBURGH, PENNA.

1910 Cochran Road
(412) 563-3313

BALA CYNWYD, PENNA.

1 Decker Square
(215) 667-1715

ATLANTA, GEORGIA

1800 Peachtree Center
230 Peachtree Street N.W.
(404) 524-8011
TWX: 810 751-8458

CORAL GABLES, FLORIDA

299 Alhambra Circle
Suite 309
(305) 443-2613
TWX: 810 848-8231

CHICAGO, ILLINOIS

O'Hare Plaza
8501 West Higgins Road
(312) 693-2087
TWX: 910 253-1826

SOUTHFIELD, MICHIGAN

18000 West Eight Mile Road
(313) 358-4260
TWX: 810 232-4966

MINNEAPOLIS, MINNESOTA

371 Tyrol West Bldg.
Gamble Center
(612) 545-0294

CINCINNATI, OHIO

Suite 2805 Atkinson Square
11750 Chesterdale Road
(513) 771-2615
TWX: 810 464-8189

HOUSTON, TEXAS

Cullen Center
500 Jefferson Bldg.
(713) 228-5321
TWX: 910 881-3731

DALLAS, TEXAS

2355 Stemmons Bldg.
(214) 637-3704
TWX: 910 861-4074

LOS ANGELES, CALIFORNIA

9841 Airport Boulevard
Suite 514
(213) 776-6080
TWX: 910 328-6527

LOS ALTOS, CALIFORNIA

885 North San Antonio Road
(415) 948-9010
TWX: 910 370-7450

WASHINGTON, D.C.

Blake Bldg. Suite 500
1025 Connecticut Avenue N.W.
(202) 659-8505
TWX: 710 822-0017

ONTARIO, CANADA

110 Yonge Street
Toronto 1, Ontario, Canada
(416) 363-2433
TWX: 610 491-1419

ENGLAND

P.O. Box 14
Burymead Road
Hitchin, Herts, England
Telephone: Hitchin 5-1185



© T.M. Sanders Associates, Inc.