

UNIVAC

UNISCOPE 100

COMMUNICATIONS  
CONTROL PROCEDURES  
UP-7779.1

UNIVAC SYSTEMS PROGRAMMING LIBRARY SERVICES

RELEASE

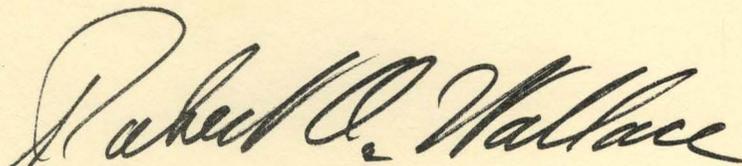
This UNISCOPE 100 Programming Information Exchange Bulletin 1, page i, announces the release and availability of "UNISCOPE 100 Display Terminal Communications Control Procedures," UP-7779.1, cover and 35 pages. This is a Restricted Distribution (RD) Item, since all Customers do not require it. Order where necessary.

One of the purposes of Programming Information Exchange Bulletins is to provide interim as well as updating materials to released publications. P.I.E. Bulletins have form numbers and may be ordered with any accompanying attachments by their "UP" number, as this one UP-7779.1.

The Communications Control Procedures comprise the data communication formats and the data interchange system which control the flow of information between the UNISCOPE 100 Display Terminal and a computer. The data communication formats, code, and code extension features conform to the American National Standard Institute (ANSI, formerly USASI) standards, and are defined in this document.

The following items are described: message formats; address and routing techniques; supervisory sequences; text format; code definitions; and line control rules.

Distribution has been made as indicated below. Additional copies may be ordered via Sales Help Requisition through your local Univac Manager from Holyoke, Massachusetts.



GROUP MANAGER  
Documentation and Library Services

**TO LISTS:** 211 (less 217),  
and S.P.L.S. Customer Lists  
37, 43, 51, 51D, 53, 53D,  
54, 54D, 55, 55D, 57, and  
60 P.I.E. Bulletin 1 only.

**ATTACHMENTS:** UP-7779.1 plus P.I.E. Bulletin 1  
to Lists 10U, 217, 630 and 692.

**THIS SHEET IS:** UNISCOPE 100  
P.I.E. Bulletin 1, UP-7779.1

**DATE:** April 13, 1970

UNISCOPE 100 DISPLAY TERMINAL

Communications Control

Procedures

This manual is published by the Univac Division of Sperry Rand Corporation as a rapid and complete means of keeping recipients apprised of UNIVAC® Systems developments. The information presented herein may not reflect the current status of the programming effort. For the current status of the programming, contact your local Univac Representative.

The Univac Division reserves the right to make such additions, corrections, and/or deletions as in the judgment of the Univac Division are required by the development of its respective Systems.

UNIVAC is a registered trademark of Sperry Rand Corporation.

Other trademarks of Sperry Rand Corporation appearing in the text of this publication are:

UNISCOPE

CONTENTS	Page
	i
1. INTRODUCTION	1
1.1. <u>FUNCTION OF EQUIPMENT</u>	1
1.2. <u>NATURE OF MANUAL</u>	1
1.3. <u>MESSAGE FORMATS</u>	1
2. ADDRESS AND ROUTING	4
2.1. <u>REMOTE IDENTIFIER - RID</u>	4
2.2. <u>STATION IDENTIFIER - SID</u>	6
2.3. <u>DEVICE IDENTIFIER - DID</u>	6
2.4. <u>GENERAL IDENTIFIER - GID</u>	6
2.5. <u>END OF ADDRESS - EOA</u>	6
3. SUPERVISORY SEQUENCES	7
3.1. <u>POLLING</u>	7
3.1.1. TEXT POLL	7
3.1.2. STATUS POLL - ENQ	10
3.1.3. RETRANSMIT REQUEST - DLE NAK	10
3.1.4. REPLY REQUEST - DLE ENQ	10
3.2. <u>TEXT (MESSAGE)</u>	10
3.3. <u>CONTROL CODES AND SEQUENCES</u>	10
3.4. <u>NO TRAFFIC - EOT EOT or EOT EOT ETX BCC</u>	13
3.5. <u>BUSY (WABT) - DLE?</u>	13
3.6. <u>BREAK AND RESUME (EMBEDDED MESSAGES) - DLE[, DLE]</u>	13
3.7. <u>NOT BUSY - DLE;</u>	13
4. TEXT FORMAT	14
4.1. <u>CURSOR ADDRESS</u>	15
4.2. <u>TEXT</u>	15
4.2.1. DATA FORMAT, COMPUTER TO UNISCOPE 100 TERMINAL	18
4.2.2. DATA FORMAT, UNISCOPE 100 TERMINAL TO COMPUTER	19
4.3. <u>ROLL AND SCROLL</u>	20
4.4. <u>LOCK KEYBOARD</u>	20
4.5. <u>TAB STOP SET</u>	20
4.6. <u>CHARACTER STRUCTURE AND BIT SEQUENCING</u>	20
4.7. <u>SYNCHRONOUS TRANSMISSION</u>	20
4.8. <u>ASYNCHRONOUS TRANSMISSION</u>	21
4.9. <u>PARITY</u>	21

5. CODE DEFINITIONS	23
5.1. <u>COMMUNICATIONS CONTROL CHARACTERS</u>	23
5.1.1. STANDARD CHARACTERS	23
5.1.2. CONTROL SEQUENCES REPRESENTED BY CODE EXTENSION	24
5.1.3. MISCELLANEOUS CONTROL CHARACTERS OR SEQUENCES	26
5.2. <u>DATA CHARACTERS</u>	27
5.3. <u>EDITING AND DEVICE CONTROL CODES</u>	27
6. MESSAGE FORMAT	31
6.1. <u>HEADING MESSAGE</u>	31
6.2. <u>COMPOSITE MESSAGE</u>	31
APPENDIX A. LINE CONTROL RULES FOR ANY SINGLE STATION OR MULTIPLEXER	32
FIGURES	
1-1. UNISCOPE 100 Display Terminal, Typical System Configuration	2
1-2. Message Formats	3
2-1. ANSCII Code Chart and UNISCOPE 100 Terminal Application	5
3-1. Communications Control Characters	7
3-2. Typical Messages Containing Supervisory Sequences	8
3-3. Status and Text Poll Message Formats	9
3-4. Reply Request Message Formats	11
3-5. Control Codes and Control Code Sequences	12
4-1. Text Format, Computer to UNISCOPE 100 Terminal	14
4-2. Text Format, UNISCOPE 100 Terminal to Computer	15
4-3. Horizontal and Vertical Cursor Addressing for UNISCOPE 100 Screen	16
4-4. Cursor Addressing for Screen Display	17
4-5. Data Format, Computer to UNISCOPE 100 Terminal	18
4-6. Data Format, UNISCOPE 100 Terminal to Computer	19
A-1. UNISCOPE 100 Terminal Traffic Flow Diagram	33

## 1. INTRODUCTION

### 1.1. FUNCTION OF EQUIPMENT

The UNISCOPE 100 Display Terminal is a low-cost, alphanumeric display that can be used for a broad range of applications requiring direct operator interaction with a central computer system. The UNISCOPE 100 terminal can operate as an entry device or as a display device.

A number of UNISCOPE 100 Display Terminals may be connected to a single communication line or to a computer input/output channel by means of a multiplexer. Additional terminals may be accommodated by cascading multiplexers. A general-purpose multiplexer is available with all the communication line interfaces on the UNISCOPE 100 Display Terminal; this permits mixing single units and multiple units on one communication system. The multiplexer also permits transmission of output messages to multiple devices. Figure 1-1 illustrates a typical system configuration.

### 1.2. NATURE OF MANUAL

This manual is a preliminary release of the Communications Control Procedures; it contains a description of the UNISCOPE 100 message address and routing of control characters, and text/message formats. Definitions of American National Standards Institute (ANSI, formerly USASI) terms and codes related to the application of the UNISCOPE 100 Display Terminal are also included. Familiarity with UNISCOPE 100 Display Terminal, General Description, UP-7701 (current version) is assumed.

### 1.3. MESSAGE FORMATS

A complete acquaintance with proper formats of messages is necessary for efficient use of this equipment. Figure 1-2 gives the general format. Three-letter groups other than those described in Figure 1-2 are fully explained in the text at or near their first appearance.

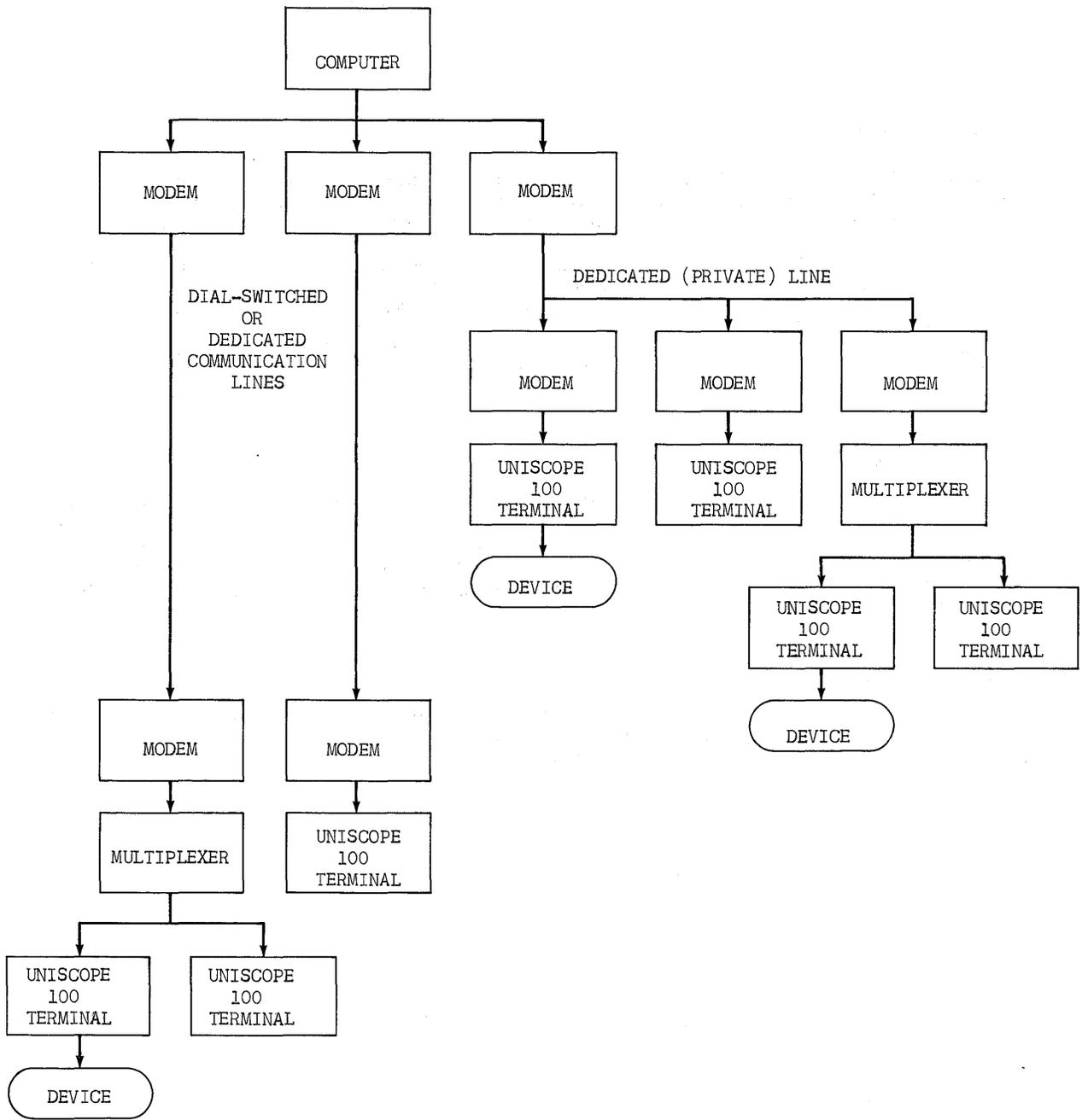
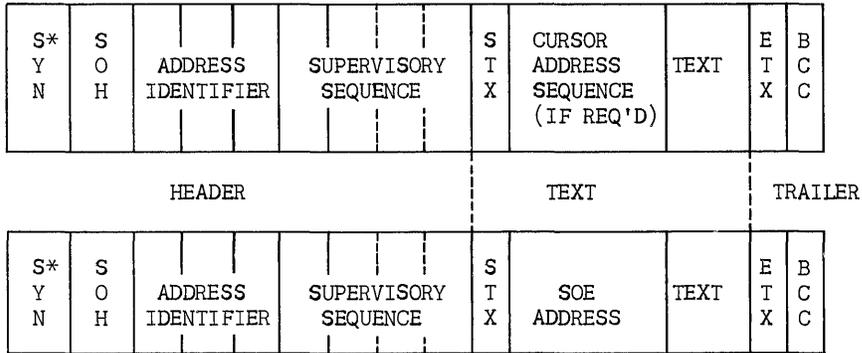


Figure 1-1. UNISCOPE 100 Display Terminal, Typical System Configuration

CENTRAL PROCESSOR TO UNISCOPE TERMINAL



UNISCOPE TERMINAL TO CENTRAL PROCESSOR

\*Four SYN Characters minimum with synchronous transmission systems only.

DEFINITIONS OF CONTROL CHARACTERS AND MESSAGE ABBREVIATIONS

BCC	BLOCK CHECK CHARACTER (PARITY CHECK)
ETX**	END OF TEXT
SOE	START OF ENTRY
SOH**	START OF HEADER
STX**	START OF TEXT
SYN**	SYNCHRONOUS IDLE

\*\*Actual transmitted characters.

Figure 1-2. Message Formats

## 2. ADDRESS AND ROUTING

The header address immediately follows the SOH character. Three characters form the address with a standard format of RID, SID, and DID. ANSCII\* characters (Figure 2-1; columns 2 through 7) are used for the address, and the column identifiers are used to distinguish addresses. An address is included in all messages except the No Traffic response from the terminal to the computer.

A specific address includes an RID, an SID, then either a DID or GID, depending upon the terminal. A general address contains a GID in either or both of the initial (RID and SID) positions, followed by a DID or GID.

### 2.1. REMOTE IDENTIFIER - RID

The RID is used for the first level of addressing. In a system using multiplexers, the RID is used to address groups of displays connected to one multiplexer. A RID must be used whether a multiplexer is used or not. The RIDs are selected from the 48 characters, SP through O, in columns 2, 3, and 4 of the code chart (Figure 2-1). The first character, SP (Space), is assigned a nonspecific function called a General Identifier (GID). If the GID is used in the RID position, any connected device would recognize it as the RID part of its address and continue reading the next two characters. The GID should not be used in the RID position if there are two or more multiplexers connected to the communications line since more than one terminal might respond, with indeterminate results. The GID in the RID position is useful on dial network circuits as part of a who-are-you message. The responding terminal never uses the GID, but always uses its own address, allowing the computer to establish its identity.

\*ANSSCII (formerly USASCI1): "American National Standard Code for Information Interchange."

BIT	b7 → b6 → b5 →					0	0	0	0	1	1	1	1
	b4	b3	b2	b1	col row	0	1	2	3	4	5	6	7
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(	8	H	X	h	x
	1	0	0	1	9	HT	EM	)	9	I	Y	i	y
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FS	,	<	L	\	l	!
	1	1	0	1	13	CR	GS	-	=	M	]	m	}
	1	1	1	0	14	SO	RS	.	>	N	^	n	~
	1	1	1	1	15	SI	US	/	?	O	_	o	(See Note 2)

RID

SID

DID

64 CHARACTER SET

LOWER  
CASE

96 CHARACTER SET

Note 1. The SP, P, and p characters constitute the GID group.

Note 2. The overpunch is not used as a delete (DEL) in the operation of the UNISCOPE 100 Display Terminal. Its bit configurations, 1111111, is utilized for an additional symbol, %.

Figure 2-1. ANSCII Character Chart and UNISCOPE 100 Terminal Application

## 2.2. STATION IDENTIFIER - SID

The SID is used for the second level of addressing. The SID specifies a terminal or station connected to the communications line by appropriate facilities either directly or through a multiplexer. The SID is selected from the 32 characters (P through 0) of columns 5 and 6 of the chart (Figure 2-1). The first character, P, is the General Identifier (GID), similar to SP, as described in 2.1. When used with an RID specifying a multiplexer, the GID in the SID position allows any station on that multiplexer to recognize its address through the SID, thereby permitting the computer to address all of the stations on a multiplexer, and also permitting the multiplexer to resolve the conflict if more than one station has traffic. This significantly reduces the polling required, by allowing one poll to query multiple terminals. The GID and the SID position is useful on dial network circuits as a part of a who-are-you message. The responding terminal uses its own address in the reply, never the GID character.

## 2.3. DEVICE IDENTIFIER - DID

The DID is used for the third and last level of addressing. The DID specifies a particular device at a terminal or station. The DID is selected from column 7 of the chart (Figure 2-1). There are 16 coded characters, from p through DEL (delete) in column 7; however, the DEL code (all 1's, the overpunch code for paper tape or cards) is not used for this application. The first code, p, is the GID character for this level, as described above. This allows the computer to address the station without having to address a device. The GID in the DID position is useful on dial network circuits as a part of a who-are-you message.

## 2.4. GENERAL IDENTIFIER - GID

The use of the General Identifier, GID, is explained in 2.1, 2.2, 2.3; however, a summary may be useful.

The address has three character positions, corresponding to the three levels of addressing. The device at the first level recognizes two characters in the RID positions, either the GID (which is SP character), or a character specifically assigned to the device. Following this, the device at the second level recognizes two characters in the SID position, again either the GID (in this case P) or a specific character; and similarly for the third level (DID position; the GID in this case is p). The address of SP, P, p is recognized by any terminal in the system.

## 2.5. END OF ADDRESS - EOA

Address characters are only those found in columns 2 through 7 of the ANSCII chart (Figure 2-1). Any character from columns 0 or 1 which follows SOH in the format, ends the address field.

### 3. SUPERVISORY SEQUENCES

A Supervisory Sequence is a sequence of one or more communication control characters and, possibly, other characters that perform a defined control function. The sequence of characters follows immediately after the last address character, DID. The communications control characters and sequences are shown in Figure 3-1. The message format with the supervisory sequences is shown in Figure 3-2.

#### 3.1. POLLING

Polling is used to solicit Text, Status, or Reply Request. If an error is detected in a polling sequence, no response is sent from any terminal in the network. A terminal never transmits except when polled. All responses except the No Traffic response are identified with the transmitting station's address in the heading.

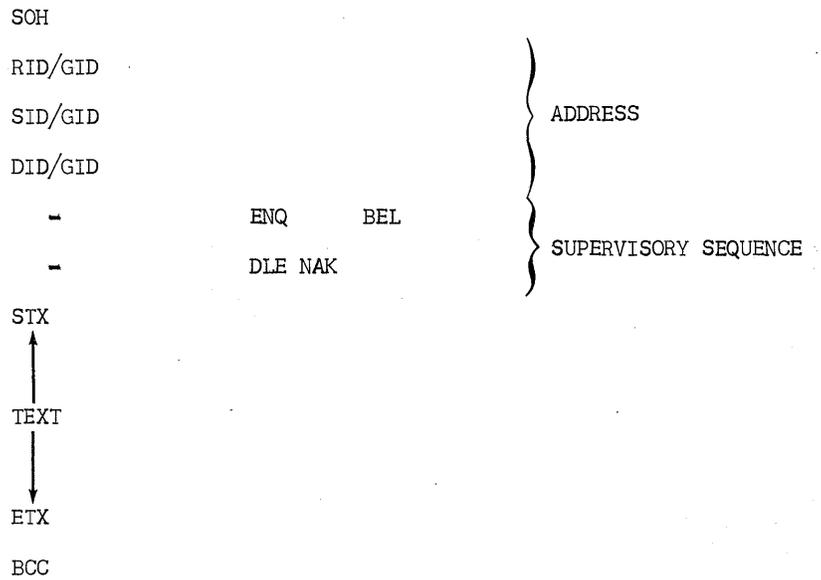
##### 3.1.1. TEXT POLL

The text poll solicits traffic from the remote terminal. Included in the response will be any outstanding acknowledgement from the addressed terminal or from any other terminal on the same multiplexer. A text poll may have either a specific or a general address (any station on a specific multiplexer). The text poll is distinguished by the lack of any other control sequence except an acknowledge sequence (if required). A message containing text cannot poll the remote terminal. Text poll formats are shown in Figure 3-3.

<u>FUNCTION</u>	<u>SYMBOL FROM COMPUTER</u>	<u>SYMBOL TO COMPUTER</u>
Start of Header	SOH	SOH
Start of Text	STX	STX
End of Text	ETX	ETX
Enquiry (Status Poll)	ENQ	-
Enquiry (Reply Request)	-	DLE ENQ
Synchronizing Character	SYN	SYN
Acknowledge	DLE 1	DLE 1
Retransmit Request	DLE NAK	-
Break	DLE[	-
Resume	DLE]	-
WABT (Busy)	-	DLE?
Traffic Response to Status Poll	-	DLE0
Not Busy	-	DLE;
Computer Message Waiting	BEL	-

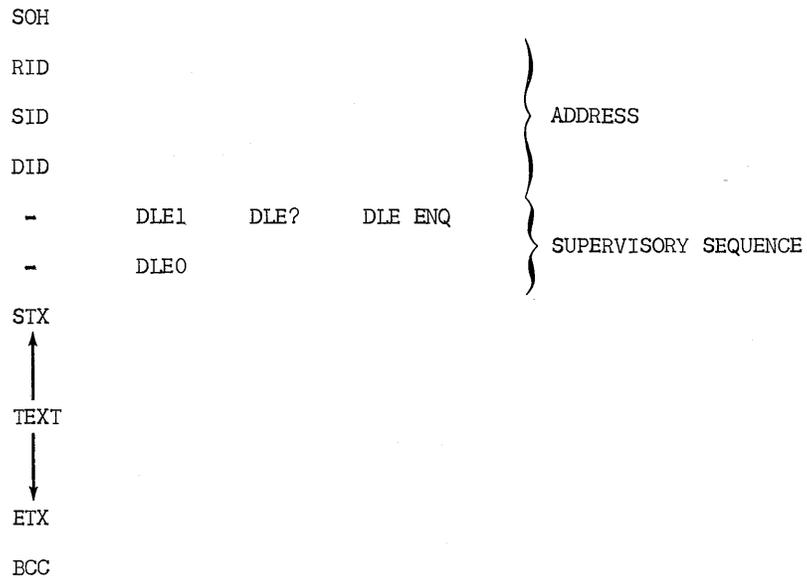
Figure 3-1. Communications Control Characters

COMPUTER-TO-UNISCOPE 100 TERMINAL



UNISCOPE 100 TERMINAL-TO-COMPUTER

POSITIVE RESPONSE



NO TRAFFIC - NO ACKNOWLEDGE

EOT or EOT  
EOT

EOT or ETX  
BCC\*

\*See 3.4

Figure 3-2. Typical Messages Containing Supervisory Sequences

COMPUTER-TO-UNISCOPE 100 TERMINAL

NOTE: In the following message formats, S/G indicates SID or GID and D/G indicates DID or GID

TRAFFIC POLL WITHOUT ACKNOWLEDGE

```
S R S D E B
O I / / T C
H D G G X C
```

TRAFFIC POLL WITH ACKNOWLEDGE

```
S R S D D E B
O I / / L 1 T C
H D G G E X C
```

STATUS POLL WITHOUT ACKNOWLEDGE

```
S R S D E E B
O I / / N T C
H D G G Q X C
```

STATUS POLL WITH ACKNOWLEDGE

```
S R S D E D E B
O I / / N L 1 T C
H D G G Q E X C
```

TEXT

```
S R S D S           E B
O I I I T ← TEXT → T C
H D D D X           X C
```

COMPUTER MESSAGE WAITING

```
S R S D B S E B
O I I I E T T C
H D D D L X X C
```

BREAK - RESUME

```
  X X D S           B D X X
... X X L [O  --- COMPLETE FORMAT --- C L] X X ...
  X X E H           C E X X
```

RETRANSMISSION REQUEST

```
S R S D D N E B
O I I I L A T C
H D D D E K X C
```

Figure 3-3. Status and Text Poll Message Formats

### 3.1.2. STATUS POLL - ENQ

The Status Poll solicits status information from the remote terminal. The response indicates if the terminal (or any terminal on a multiplexer if a general address is used) has any traffic. The traffic is not sent, but is held for a traffic poll. The DLE0 sequence in the response indicates that the terminal has traffic. The response includes any outstanding acknowledgement from the addressed terminal and from any other terminal on the same multiplexer. A Status Poll may have either a specific or a general address. The ENQ character is used to solicit status. Status Poll formats are shown in Figure 3-3. The Status Poll may bring in a Request Computer Message response.

### 3.1.3. RETRANSMIT REQUEST - DLE NAK

The Retransmit Request from the computer causes the terminal to retransmit its last response. The Retransmit Request is sent in response to the Reply Request (DLE?) from the terminal, indicating that the text sent to the computer has not been acknowledged. If the text was sent, an acknowledgement is sent, or repeated, instead of a Retransmission Request. Retransmission Request formats are shown in Figure 3-3.

### 3.1.4. REPLY REQUEST - DLE ENQ

The Reply Request solicits the acknowledgement for the last transmission. If a terminal does not receive an acknowledgement for text sent to the computer with the next poll recognized by the transmitting terminal, the terminal sends a Reply Request to the computer. This allows the computer to determine whether the text or the acknowledgement for the text was lost, and to send an acknowledgement or cause the terminal to retransmit the text. The terminal will continue to respond to its polls with a Reply Request until the computer either acknowledges the text or requests a retransmission. The DLE ENQ sequence solicits the reply. Reply Request formats are shown in Figure 3-4. After missing an acknowledgement, the terminal will accept only an acknowledgement having a specific address.

## 3.2. TEXT (MESSAGE)

The text message is a sequence of characters which conveys the text and the information necessary to handle the text. The text message begins with the STX character and ends with the ETX character.

```
  S           E
  T ← TEXT → T
  X           X
```

## 3.3. CONTROL CODES AND SEQUENCES

The control codes and control code sequences used with the UNISCOPE 100 terminal are shown in Figure 3-5. Not all of the coding illustrated is necessary for programming the UNISCOPE 100 terminal, but it is included for completeness of information. A definition of the codes is included in Figure 3-5.

UNISCOPE 100 TERMINAL TO COMPUTER  
RESPONSES TO TRAFFIC AND STATUS POLLS

BUSY

```
S R S D D   E B
O I I I L ? T C
H D D D E   X C
```

NO TRAFFIC WITHOUT ACKNOWLEDGE

```
E E           or       E E E B
O           O       O O T C
T           T       T T X C
```

NO TRAFFIC WITH ACKNOWLEDGE\*

```
S R S D D   E B
O I I I L 1 T C
H D D D E   X C
```

REPLY REQUEST

```
S R S D D E E B
O I I I L N T C
H D D D E Q X C
```

REQUEST COMPUTER MESSAGE WITHOUT ACKNOWLEDGE

```
S R S D B E B
O I I I E T C
H D D D L X C
```

REQUEST COMPUTER MESSAGE WITH ACKNOWLEDGE\*

```
S R S D D   B E B
O I I I L 1 E T C
H D D D E   L X C
```

THRU (PRINTING) WITHOUT ACKNOWLEDGE

```
S R S D D   E B
O I I I L ; T C
H D D D E   X C
```

THRU (PRINTING) WITH ACKNOWLEDGE\*

```
S R S D D   D E B
O I I I L 1 L ; T C
H D D D E   E X C
```

RESPONSES TO STATUS POLL ONLY

TRAFFIC WITHOUT ACKNOWLEDGE

```
S R S D D   E B
O I I I L O T C
H D D D E   X C
```

TRAFFIC WITH ACKNOWLEDGE\*

```
S R S D D   D E B
O I I I L 1 L O T C
H D D D E   E X C
```

RESPONSES TO TRAFFIC POLL ONLY

TEXT WITHOUT ACKNOWLEDGE

```
S R S D S           E B
O I I I T ← TEXT → T C
H D D D X           X C
```

TEXT WITH ACKNOWLEDGE\*

```
S R S D D   S           E B
O I I I L 1 T ← TEXT → T C
H D D D E   X           X C
```

\*Busy (DLE?) may be sent instead of an acknowledgement (DLE1) by a terminal that has received text correctly but cannot accept more text immediately. A specifically addressed poll to a unit that is busy will always bring the Busy response. The Busy response will be returned only once to a general poll.

Figure 3-4. Reply Request Message Formats

<u>FUNCTION</u>	<u>CODE FROM COMPUTER</u>	<u>CODE TO COMPUTER</u>
Carriage Return (New Line)	CR ①	CR
Erase to End of Display	ESC a	---
Erase to End of Line	ESC b	---
Delete - In Line	ESC c	---
Delete - In Display	ESC C	---
Insert - In Line	ESC d	---
Insert - In Display	ESC D	---
Scan Left	ESC g	---
Scan Right	ESC h	---
Scan Down	ESC i	---
Scan Up	ESC f	---
Character Erase (Space)	SP	SP
TAB	HT ②	HT
Message Waiting	BEL	---
Cursor to Home	ESC e	---
Insert Line	ESC j	---
Delete Line	ESC k	---
Request Comp. Message	---	BEL ③
TAB STOP SET	ESC HT	---
Start Blink Field	FS	FS
End Blink Field	GS	GS
Lock Keyboard	DC4	---
Print (Output)	DC2	---
Print Transparent (Output)	---	---
Start of Entry (SOE) ▷	RS	RS
SI ④	SI	SI
SO ④	SO	SO
LF ⑤	LF	LF
FF ⑤	FF	FF
EM ⑤	EM	EM
Transmit	DC1	---

NOTES:

- ① ESC      The ESC code is used to provide control sequences as defined in the proposed ANSI Standard for Code Extension Procedures for Information Interchange, published in the Communications of the ACM, December 1968, "Control Set Extension: Use of ESC (ESCAPE)."
- ② BEL      The Computer Message Waiting symbol (BEL) must be placed between the address (RID, SID, DID) and the start of text (STX). If it follows STX, it is placed in memory and does not light the Computer Message Waiting light. It is not displayed if in memory.
- ③ BEL      BEL sent to the computer in a nontext message (no STX) indicates that the Message Waiting key was depressed. If BEL appears in text, it was placed in the UNISCOPE 100 terminal either by the computer or through the parallel interface and is to be interpreted by the user program.
- ④ SI, SO    The SO character is used as a part of the cursor position sequence. Otherwise the UNISCOPE 100 terminal is transparent\* to these characters. SI and SO are reserved for use by the UNISCOPE 100 terminal for graphic set extension or by devices connected to the parallel channel as defined in the proposed ANSI Standard for Code Extension Procedures for Information Interchange published in the Communications of the ASM, December 1968, "Graphic Set Extension: Use of SO (Shift Out) and SI (Shift In)". SI is used by peripherals on parallel channels only.
- ⑤ LF, FF, EM    The UNISCOPE 100 terminal is transparent\* to these codes. They are not displayed. LF, FF may be placed in UNISCOPE 100 terminal by either the computer or the parallel interface.

\*The term "transparent" signifies that the UNISCOPE 100 terminal is unaffected by the characters referred to, but transmits them for use by other devices.

Figure 3-5. Control Codes and Control Code Sequences

3.4. NO TRAFFIC - EOT EOT or EOT EOT ETX BCC

The No Traffic response sent by a terminal indicates that the terminal has neither text nor an acknowledgement to send, and is not missing an acknowledgement for text transmitted (Reply Request). If the terminal has no traffic but has an outstanding acknowledgement to send, the normal format with address and block parity check is sent. Formats for no traffic responses are shown in Figure 3-4. The long format (EOT EOT ETX BCC) is required by the Communication Terminal Modular Control (CTMC).

3.5. BUSY (WABT) - DLE?

The Busy response by a terminal indicates that the terminal is in an extended sequence, such as printing, and is not immediately available. A Busy response is returned instead of an acknowledgement when a terminal receives error-free text and cannot accept more text immediately. The treatment of a Busy response is similar to that of an acknowledgement, with respect to format and inclusion in messages from other terminals on a common multiplexer. The Busy response will not be sent in response to subsequent general polls after the first Busy response sent as an acknowledgement to the initial general poll. Busy formats are shown in Figure 3-4.

3.6. BREAK AND RESUME (EMBEDDED MESSAGES) - DLE[, DLE]

Messages may be embedded in other messages by delimiting the embedded message with the DLE[ and DLE] characters. The embedded message must not be addressed to the same terminal as the broken message. The embedded message may not be broken (only one level of embedding). The DLE[ and DLE] characters are not included in the block check of either the broken or the embedded message. The embedded message has an independent block check character (BCC). The DLE] character initiates the resumption of the previous message. The break must occur between STX and ETX; it must not occur between ETX and BCC.

3.7. NOT BUSY - DLE;

The Not Busy sent from the terminal indicates that the terminal has completed an extended sequence, and is now available for further traffic. The Not Busy response must be used only subsequently to a Busy response.

#### 4. TEXT FORMAT

The formats of text between the computer and the UNISCOPE 100 terminal are shown in Figures 4-1 and 4-2. In transmissions from the computer to the UNISCOPE 100 terminal, the text is placed on the screen of the UNISCOPE 100 terminal, starting at the position of the cursor. If the cursor is not repositioned after data has been transmitted to the computer, text received for the computer will follow the text already displayed on the screen. The cursor may be positioned by several different methods, before or during transmission.

- The cursor address sequence positions the cursor to any specified point on the screen.
- "Cursor-to-home" sequence places the cursor in the upper left hand corner of the screen.
- The carriage return and space characters position the cursor relative to its last position.
- The TAB character moves the cursor in the same manner as the TAB key on the keyboard of a typewriter.

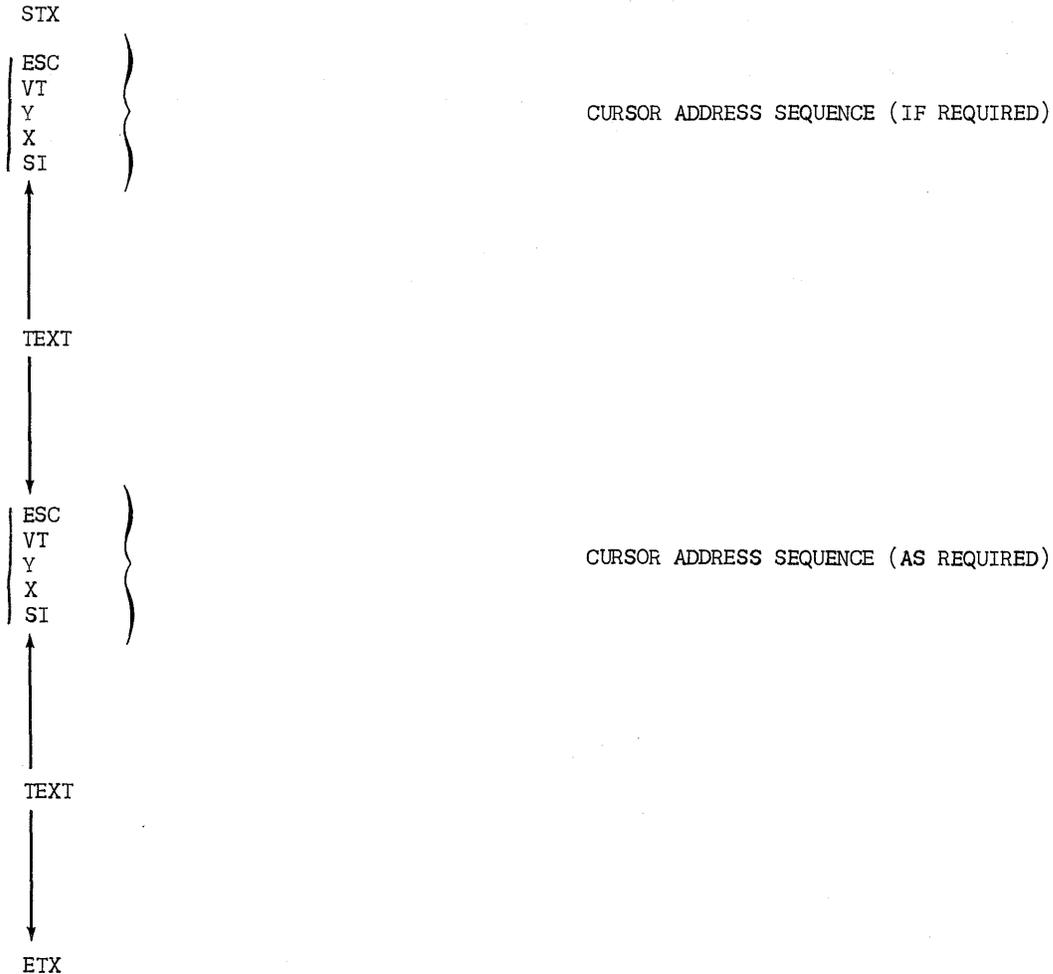


Figure 4-1. Text Format, Computer to UNISCOPE 100 Terminal

#### 4.1. CURSOR ADDRESS

The format for the cursor address in a text message from the computer to the UNISCOPE 100 terminal is the first one shown in Figure 4-1.

The ESC (Escape) character determines the meaning of the following character to permit extended control sequences. The VT character following ESC denotes that the next two characters are the cursor address; the first of the two identifies the row, and the second the column, according to the character assignments given in Figures 4-3 and 4-4.

Figure 4-3 lists the address coding. Columns 0 and 1 of the address code table are not used; these are for control. Column 7 is not used because no address greater than 80 is required with the UNISCOPE 100 terminal. The exact code characters chosen are a function of the number of rows and columns in the particular UNISCOPE 100 terminal being addressed and the desired position on the screen. The SI character is the character specified by the ANSCII code to end the sequence started by the ESC VT codes. The cursor address sequence may be placed anywhere in text after the STX character and before the ETX character. There is no limit to the number of times the cursor address sequence may be used in one message or on the position that may be addressed.

#### 4.2. TEXT

The text sent from the computer to the UNISCOPE 100 terminal starts at the cursor position and moves to the right across the screen, superseding character by character any prior text displayed. A carriage return character (CR) in the text will cause the next character to appear at the beginning of the next line. If the end of a line is reached, the next character will appear at the beginning of the next line, without the need for a carriage return character. If a carriage return character is used before the end of a line, the rest of the line is unaffected. The Erase to End of Line sequence (ESC b) may be used to erase the rest of the line. The Erase to End of Display sequence (ESC a) may be used to clear the remainder of the screen, following the end of the message if desired. It is recommended that the Erase to End of Display sequence be used at the end of a message rather than at the beginning of a message to avoid delay that would occur while the erasure is being accomplished.

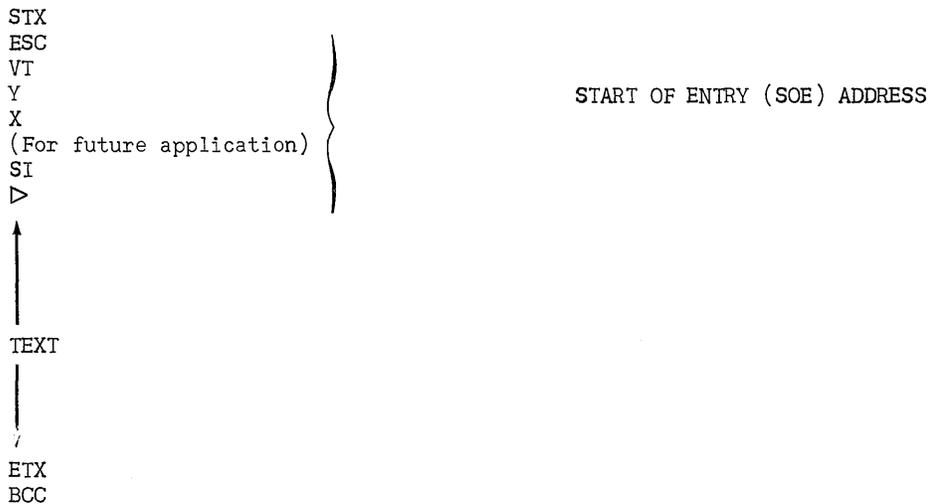


Figure 4-2. Text Format, UNISCOPE 100 Terminal to Computer

					0	0	0	0	1	1	1	1
					0	0	1	1	0	0	1	1
					0	1	0	1	0	1	0	1
b4 ↓	b3 ↓	b2 ↓	b1 ↓	col row	0	1	2	3	4	5	6	7
0	0	0	0	0			1	17	33	49	65	
0	0	0	1	1			2	18	34	50	66	
0	0	1	0	2			3	19	35	51	67	
0	0	1	1	3			4	20	36	52	68	
0	1	0	0	4			5	21	37	53	69	
0	1	0	1	5			6	22	38	54	70	
0	1	1	0	6			7	23	30	55	71	
0	1	1	1	7			8	24	40	56	72	
1	0	0	0	8			9	25	41	57	73	
1	0	0	1	9			10	26	42	58	74	
1	0	1	0	10			11	27	43	59	75	
1	0	1	1	11			12	28	44	60	76	
1	1	0	0	12			13	29	45	61	77	
1	1	0	1	13			14	30	46	62	78	
1	1	1	0	14			15	31	47	63	79	
1	1	1	1	15			16	32	48	64	80	

Figure 4-3. Horizontal and Vertical Cursor Addressing for UNISCOPE 100 Screen



4.2.1. DATA FORMAT, COMPUTER TO UNISCOPE 100 TERMINAL

An example of the data format used in transmission from the computer to the UNISCOPE 100 terminal is shown in Figure 4-5. In this example, the phrase "The Quick Brown Fox Jumps Over", is transmitted to the UNISCOPE 100 terminal but by use of the cursor addressing sequence and the carriage return, the data is placed on the screen as shown in the right of Figure 4-5. The initial cursor position places the first letter on the second line in the fifth position. The rest of the first two words are placed on the screen in sequence, the second cursor positioning sequence places the first letter of the third word on the fifth line and in the ninth character position. The rest of the word follows in sequence. The carriage return at the end of the word causes the first letter of the fourth word to be in the first character position of the next line. The two carriage returns followed by two spaces at the end of the fourth word causes the first letter of the fifth word to be in the second position of the eighth line. The third cursor position sequence causes the first letter of the last word to be in the first character position of the first line returning the cursor to the beginning of the screen. Since no erase sequences were used, the screen remains as before the transmission except for the six words transmitted and the space between the first two words and the spaces before the fifth word. This allows the computer to change specific characters, words, phrases, sentences, or paragraphs on the screen without disturbing or repeating the data that is already on the screen.

```

STX
ESC
VT
!           1 2 3 4 5 6 7 8 9 10 11 12 13
$
SI          1 - O v e r
T           2 -           T h e   Q u i c k
h           3 -
e           4 -
SP          5 -           B r o w n
Q           6 - F o x
u           7 -
i           8 -   J u m p s
c
k

```

```

ESC }
VT  } 1
$   }
(   }
SI  }
B
r
o
w
n
CR
F
o
x
CR }
CR } 2
SP }
SP }
J
u
m
p
s
ESC }
VT  } 3
SP }
SP }
SI  }
O
v
e
r
ETX

```

RESULT ON UNISCOPE 100 SCREEN ILLUSTRATING:

- (1) Cursor Addressing
- (2) Use of Carriage Return
- (3) Cursor Addressing in Text

Figure 4-5. Data Format, Computer to UNISCOPE 100 Terminal

4.2.2. DATA FORMAT, UNISCOPE 100 TERMINAL TO COMPUTER

The format for data from the UNISCOPE 100 terminal to the computer is shown in Figure 4-6. In this example, the transmit key is depressed while the text "The Quick Brown" shown at the top of the figure is on the screen. The data that is transmitted is the portion between the start of entry (SOE) character,▷, closest to the cursor and the cursor (|). Immediately following STX is an Escape sequence (ESC VT) giving the address of the SOE. The two characters following the ESC VT are the row and column positions of the SOE character. The third character following the ESC VT sequence is reserved for future application and is not used at this time. The SI character is the ANSCII symbol to end the address sequence. Immediately following the address sequence, the first text character follows the SOE. The text continues to the last significant character, in this case, n. Nonsignificant spaces that follow are suppressed, and a carriage return (CR) character is inserted. The first character on the next line follows the carriage return. Spaces between words are included. Again nonsignificant spaces after the s in "Jumps" are suppressed, and a carriage return follows the s. On the last line, the spaces are included between the last significant character, e, and the cursor. By transmitting the address of the SOE character at the beginning of the text and also spaces out to the cursor, the computer program can accurately ascertain the location of the transmitted text on the face of the screen. Since the computer can control the unlocking of the keyboard, it can position the next reply to the terminal at the end of the data transmitted from the terminal.

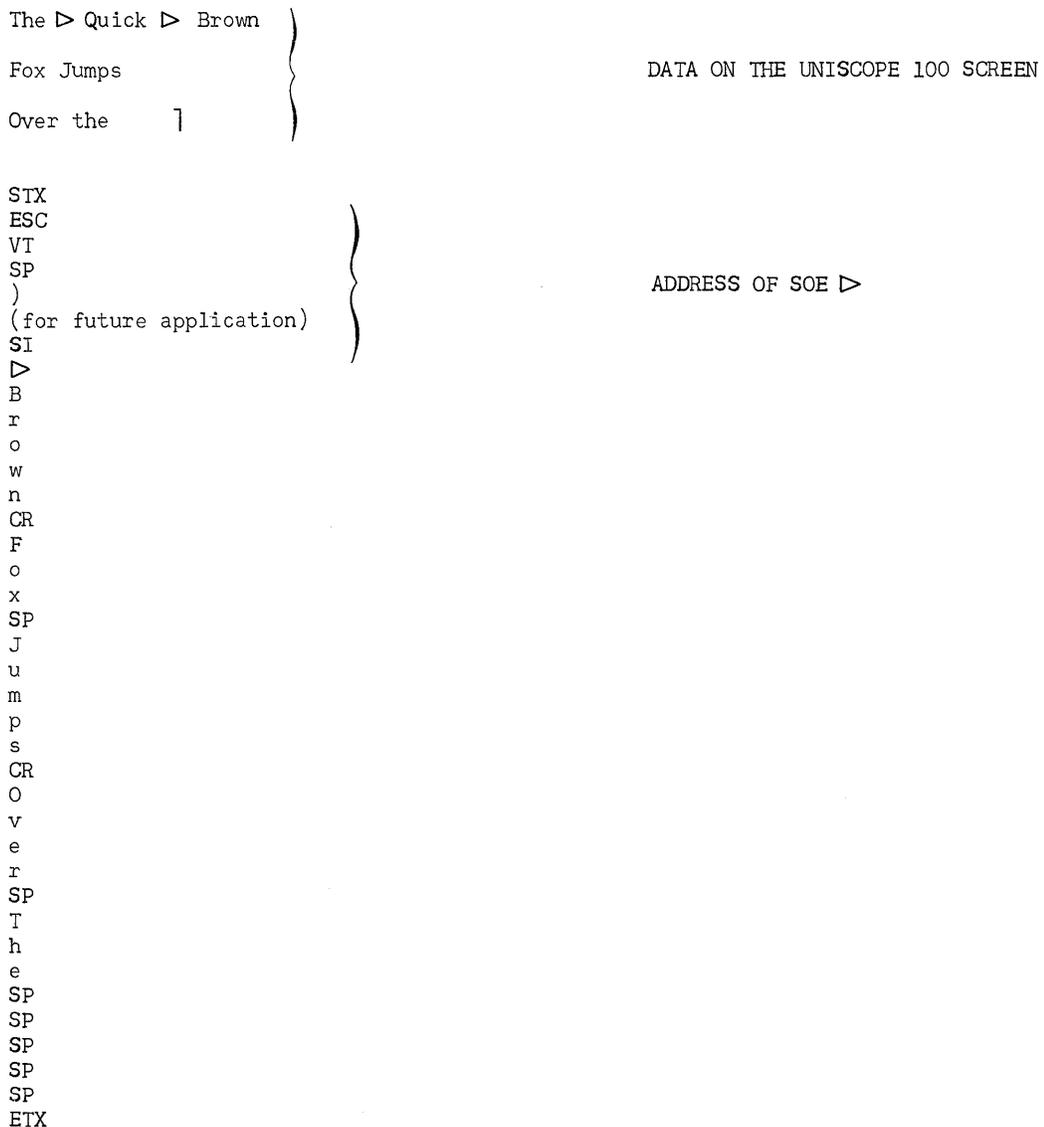


Figure 4-6. Data Format, UNISCOPE 100 Terminal to Computer

Since the computer can include SOE in its output text, it can insert the required SOE for the operator, thereby simplifying the operational sequence. By having the SOE character positioned at the end of a question, the operator can simply type in the answer and press the transmit key without being concerned about either the SOE or the position of the data on the screen.

#### 4.3. ROLL AND SCROLL

The effect of the data rolling downward or upward across the face of the screen can be achieved by the use of the line insert (ESC j) or line delete sequences (ESC k), respectively.

The line insert function causes the contents of the line at and below the cursor position to move down one line, the contents of the last line being lost; the result is a blank line inserted at the cursor position. By transmitting a single line of data into the cursor position, a line has been inserted.

When the line delete function is transmitted, the contents of the line at the cursor position are lost and the lines below that line are all moved up to close the space; the bottom line is then blank. By repositioning the cursor to the bottom line and transmitting a single line of data the selected portion of the screen has been moved up one line. By repeating this, with appropriate timing, the contents of the screen will move either up or down.

#### 4.4. LOCK KEYBOARD

Each text transmission from the computer to the display locks the keyboard at the beginning of the text. At the end of the text, the keyboard is unlocked unless the keyboard lock character (DC4) is included in the text. This gives the computer control over the keyboard.

#### 4.5. TAB STOP SET

The computer can insert tab stops (analogous to a typewriter) in text transmitted to the UNISCOPE 100 terminal. The sequence ESC HT is inserted at each point in the text where a tab stop is desired. When the operator presses the TAB key, the cursor moves to the next tab stop in sequence. This can be repeated until the cursor reaches the end of the screen, where it stops. This makes it convenient for the operator to fill in blanks in forms. The character HT is placed in memory as the tab stop.

#### 4.6. CHARACTER STRUCTURE AND BIT SEQUENCING

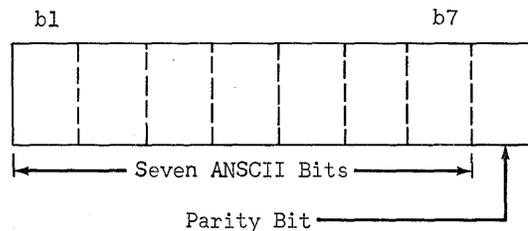
The character structure conforms to ANSI X3.16 - 1966 Character Structure and Character Parity Sense for Serial-by-Bit Data Communications in the USA Standard Code for Information Interchange.

The bit sequencing conforms to ANSI X3.15 "Bit Sequencing of the USA Standard Code for Information Interchange in Serial-by-Bit Data Transmission".

#### 4.7. SYNCHRONOUS TRANSMISSION

The character structure for synchronous data communications consist of eight bits; there are seven ANSCII bits plus one character parity bit.

The bit sequence for an ANSCII character is least significant bit, b1, first, to most significant bit, b7, in terms of the ANSCII nomenclature, (ANSI X3.4 "Code for Information Interchange") in ascending order.



NOTE: The order of transmission is from left to right.

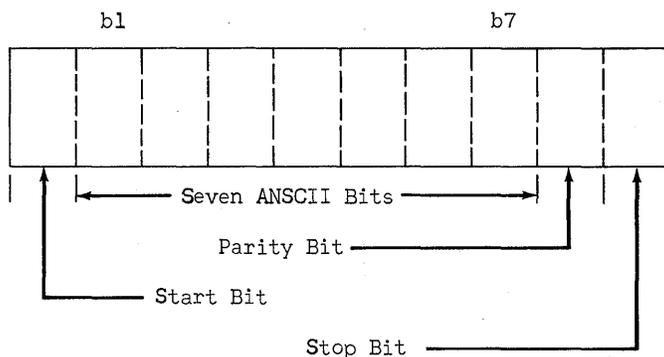
The UNISCOPE 100 Terminal generates a parity bit and adds it to every seven-bit code transmitted. When receiving, it checks the character parity. The character parity for synchronous transmission is odd; that is, there is an odd number of 1 bits per character.

#### 4.8. ASYNCHRONOUS TRANSMISSION

The character structure for asynchronous data communication consists of 10 signal elements having equal time intervals; one 0 (spacing) start element, seven ANSCII bits, one character parity bit, and one 1 (marking) stop element. The intercharacter interval (the time interval between the end of a stop element and the beginning of the next start element) may be of any length, and is of the same sense as the stop (marking) element, that is, 1.

The bit sequence for an ANSCII character shall be least significant bit, b1, first to most significant bit, b7, in terms of the ANSCII nomenclature (ANSSCII, X3.4) in ascending consecutive order.

EXAMPLE:



NOTE: The order of transmission is from left to right.

The UNISCOPE 100 terminal supplies the start, parity, and stop bits to the seven ANSCII bits on transmission. On receiving, it uses the start and stop bits to establish bit and character timing, checks parity, and acts on the seven ANSCII bits. The character parity for asynchronous transmission is even, i.e., an even number of 1 (marking) bits per character.

#### 4.9. PARITY

Character parity conforms to ANSI X3.16 - 1966 "Character Structure and Character Parity Sense for Serial-by-Bit Data Communications in the USA Standard Code for Information Interchange".

Message Parity conforms to ANSI X3.3.4/212 "Proposed American National Standard Data Communications Control Procedures for the American National Standard Code for Information Interchange".

The sense of character parity for synchronous data and communications is odd over the eight bits; that is, there is an odd number of 1 bits per character.

The sense of character parity for asynchronous data communications is even over the eight bits (seven ANSCII bits and the character parity bit); that is, there is an even number of 1 bits per character.

The message parity character is the Block Check Character (BCC) defined in ANSI X3.3.4/212, previously cited.

During transmission, the UNISCOPE 100 terminal generates a BCC and transmits it to the computer as the last character of every message. During receiving, the UNISCOPE 100 terminal checks the BCC.

The BCC is generated by taking the binary sum independently (without carry) on each of the seven individual levels of the transmitted code (b1 to b7). If a sum is odd, a 1 bit is put into the corresponding position of the BCC. (The longitudinal parity is even.)

The character parity bit of the BCC itself is the same sense as the character parity of the text characters; the parity is even for asynchronous transmission, odd for synchronous transmission.

The BCC calculation starts immediately following SOH. All characters, except SYN, which are transmitted after SOH are included in the BCC calculation. This includes ETX, which signals that the next character following is BCC, but excludes the break and resume sequences and all characters between them.

The BCC must immediately follow the ETX character. SYN must not be transmitted between ETX and BCC. Note that the BCC may produce any of the 128 code combinations.

**EXAMPLE:**

CHARACTER PARITY

MSG	b7	b6	b5	b4	b3	b2	b1		SYNC	ASync
SOH	0	0	0	0	0	0	1		0	1
5	0	1	1	0	1	0	1		1	0
h	1	1	0	1	0	0	0		0	1
p	1	1	1	0	0	0	0		0	1
STX	0	0	0	0	0	1	0		0	1
A	1	0	0	0	0	0	1		1	0
ETX	0	0	0	0	0	1	1		1	0
BCC	1	1	0	1	1	0	0		1	0

**NOTE:** The BCC is the same (b1 through b7) for both synchronous and asynchronous transmission.

## 5. CODE DEFINITIONS

The discussion of codes is covered in three areas: (1) communications control characters used to control the communications over the communications link; (2) data or text characters used to convey the actual message; (3) function codes used to control the UNISCOPE display. Except where otherwise stated, the code used is ANSI X3.4 - 1967 "Code for Information Interchange".

### 5.1. COMMUNICATIONS CONTROL CHARACTERS

#### 5.1.1. STANDARD CHARACTERS

##### SOH - Start of Heading

SOH is a communication control character used at the beginning of a sequence of characters which constitutes a machine-sensible address or routing information. This sequence is referred to as the "heading". The heading is terminated by STX or by ETX.

##### Use

- (1) SOH marks the start of a message heading.
- (2) SOH may not deselect a station without being followed by an appropriate address and block ending.

##### STX - Start of Text

STX is a communication control character which precedes a sequence of characters that is to be transmitted in its entirety through to the ultimate destination. Such a sequence is referred to as "text". STX may be used to terminate a sequence of characters (heading) started by SOH.

##### Use

- (1) STX marks the start of a message text.
- (2) If a heading precedes the text, STX marks the end of the message heading.

##### ETX - End of Text

ETX is a communication control character used to terminate a sequence of characters started with STX or SOH and transmitted as an entity.

##### Use

- (1) ETX marks the end of a message text.
- (2) ETX may also mark the end of a heading.

##### EOT - End of Transmission

EOT is a two-character contiguous sequence consisting of EOT=EOT.

EOT is a communication control sequence used to indicate the conclusion of a transmission, which may have contained one or more texts and any associated headings. The sequence may also be used to indicate "No Traffic" in response to a poll.

##### Use

- (1) Receipt of the EOT sequence by a control station, subsequent to a poll, indicates that the polled-tributary station has no traffic to send.
- (2) The EOT sequence never immediately follows terminal-to-computer traffic when operating in interactive mode.
- (3) The EOT sequence is never transmitted from a control station to a tributary station in interactive mode.

### ENQ - Enquiry

ENQ is a communication control character used in data communication systems to request a response from a remote station. It may be used as a "who-are-you?" (WRU) to obtain identification, or may be used to obtain station status, or both.

ENQ is the basic character for the status poll.

#### Use

- (1) ENQ is used to solicit status from a station.
- (2) ENQ is not used to solicit retransmissions.

### SYN - Synchronous Idle

A communication control character used by a synchronous transmission system, in the absence of any other character, to provide a signal from which synchronism may be achieved or retained.

#### Use

- (1) SYN is used to achieve and maintain character synchronism in synchronous communication systems.
- (2) SYN is used as a communications "time-fill" character during periods in a transmission when no other characters are available to send.
- (3) SYN may be transmitted by both central and remote stations as described below.
  - After a period in which no characters have been transmitted on a channel, and prior to the transmission of any other character, at least four SYN characters must be transmitted.
  - Determination of synchronization, once achieved, and the recognition of character synchronization are the responsibility of the receiving station. No station is considered synchronized until two successive SYN characters have been received. All stations in a multipoint mode remain in synchronization when not sending, if data is present on the receive line.
  - Stations are considered out of synchronization when they detect three successive characters other than SYN, which are not control functions, while the stations are not in a data mode.
  - When the SYN character is used as a communications "time-full" character during a transmission, SYN may be arbitrarily added at any point in the transmission except:
    - (1) in a control sequence following DLE
    - (2) between ETB or ETX and the next following BCC
  - The SYN character is not to be used for time-fill or media-fill functions that are to be conveyed through the system. The receiving station deletes all SYN characters.

### DLE - Data Link Escape

DLE is a communication control character which changes the meaning of a limited number of immediately following characters. It is used only to provide supplementary controls in data communication networks.

#### Use

Additional control functions are represented by an unbroken sequence of characters, the first character of which is always DLE.

## 5.1.2. CONTROL SEQUENCES REPRESENTED BY CODE EXTENSION

### DEOT - Mandatory Disconnect

DEOT is a communication control sequence commanding the disconnection of a communication link. It comprises the character sequence DLE EOT.

### Use

- (1) DEOT indicates the end of a transmission and results in the mandatory disconnection of a circuit-switched communication link.
- (2) A master station relinquishes its right to transmit by sending DEOT.
- (3) The receipt of DEOT cancels any previous selection of a remote station.
- (4) A DEOT is never specifically addressed to only one station.
- (5) No response is made by the remote station to a DEOT.

### ACKn - Acknowledgement n

<u>Abbreviation</u>	<u>Representation</u>
ACK0	DLE followed by 0
ACK1	DLE followed by 1

ACKn is a set of communication control sequences transmitted by a receiver as affirmative responses to a sender. These are used in lieu of ACK to number affirmative replies.

### Use

- (1) ACKn is transmitted by a remote station as a numbered affirmative reply.
- (2) ACK1 is used as the acknowledgement in those links using single acknowledgements. It has all the functional properties of ACK.

### WABT - Wait Before Transmit

WABT is a communication control sequence sent by a remote station when an error free block or message has been received but the station is temporarily unable to receive more traffic. It comprises the character sequence DLE ?

### Functional Properties

WABT is transmitted by the remote station when it is not ready to receive, or when it has not completed a control function.

### DNAK - Negative Acknowledgement

The DNAK is a communication control sequence sent by a master station to indicate a negative reply. Primarily used in conjunction with ACKn, DNAK has all the functional properties of NAK. It comprises the character sequence DLE NAK.

### Function

- (1) DNAK is transmitted by a remote station as a negative reply when operating in the batch mode, or as a retransmission request for the last block or message.
- (2) DNAK is transmitted by a master station as a retransmission request when operating multipoint. It is used in response to a DENQ Reply Request from the remote station.

### DBR - Break (For Embedded Segment)

DBR is a communication control sequence used to break a message for the purpose of inserting another message to another station. It is represented by the character sequence DLE [

The DRB sequence is not included in the BCC of either message block.

### Function

- (1) It identifies the following characters as being in a new block.
- (2) It suspends the calculation of the BCC of the first block.
- (3) It causes the receiver of the first block to stand by.

### DRE - Resume (After Embedded Segment)

DRE is a communication control sequence used to terminate the insertion begun by the DBR. It is represented by the character sequence DLE].

The DRE sequence is not included in the BCC of either message block.

### Function

- (1) It marks the end of an embedded message.
- (2) It returns the interrupted receiver to its previous condition.
- (3) It initiates the resumption of the calculation of the BCC for the original block.

### DENQ - Modified Enquiry

DENQ is a communication control sequence used in data communication systems as a request for a response from a remote station. It comprises the character sequence DLE ENQ.

## 5.1.3. MISCELLANEOUS CONTROL CHARACTERS OR SEQUENCES

The miscellaneous control characters are not necessarily communication controls, but they are related and are included here as a matter of record.

### EM - End of Medium

EM is a control character associated with transmitted data; it may be used to identify the physical end of the medium or the end of the used or wanted portion of information recorded in a medium.

### Use

- (1) It is normally used in fixed block systems to inform the receiving station that the block being received has fewer characters than normal.
- (2) It may be followed by ETX or ETB or embedded block.
- (3) EM is included in the data and the redundancy check.

### NUL - Null

The NUL is an all-zeros character which may serve to accomplish time fill and media fill.

### Use

The NUL character is used for time-fill or media-fill functions that are to be conveyed through the system.

### SO - Shift Out

SO is a control character which indicates that the code combinations that follow are not to be interpreted as a member of the character set of the standard code table until a Shift In character is reached.

### Use

It causes the replacement of the standard set of graphics by an alternate set.

### SI - Shift In

SI is a control character indicating that the code combinations which follow are to be interpreted according to the standard code table.

### Use

It causes the replacement of an alternate set of graphics by the standard set.

### ESC - Escape

ESC is a control character intended to provide code extension (supplementary characters) in general information interchange. The Escape character itself is a prefix affecting the interpretation of a limited number of contiguously following ANSCII characters.

#### Use

The combination of ESC and the following characters are used to represent a control function not directly represented within the code.

### BCC - Block Check Character

BCC is a character added at the end of a message or transmission block to facilitate error detection as follows:

- (1) The BCC is generated by taking a binary sum independently (without carry) on each of the 7 individual levels of the transmitted code ( $b_1$  to  $b_7$ ).
- (2) In each code level, the number of 1 bits, including any in the BCC, is caused to be even. Thus, the sense of longitudinal parity is said to be even whether the transmission is synchronous or asynchronous.
- (3) The correct value of the character parity bit of the BCC itself is that which makes the sense of character parity the same as for text character.

#### Function

- (1) Counting for the BCC is started immediately following the first appearance in a message or transmission block of either SOH or STX. STX may appear once in a block started with SOH; in that case, the count includes STX.
- (2) All characters following the delimiting character (SOH or STX) are included in the summation except SYN characters, and embedded messages with their delimiters, DBR and DRE.

The summation includes the delimiting control character which signals that the next following character is the BCC.

- (3) The BCC is transmitted as the next character following the transmission of each end-of-text or end-of-block marker (ETX or ETB).

## 5.2. DATA CHARACTERS

The data characters are the graphic symbols taken from ANSI X3.4 - 1967, "Standard Code for Information Interchange." The character set is shown in Figure 2-1. The 64-character option includes the characters shown in columns 2 through 5. The 96-character option includes the characters in columns 2 through 7.

## 5.3. EDITING AND DEVICE CONTROL CODES

### CARRIAGE RETURN

### CR

The Carriage Return character sent from the computer to the UNISCOPE 100 terminal causes the cursor on the UNISCOPE 100 terminal display to move to the beginning of the next line. The Carriage Return is not placed in memory. Sent from the UNISCOPE 100 terminal to the computer, Carriage Returns are inserted at the end of each line (after nonsignificant spaces are suppressed) in the data transmitted to the computer.

### ERASE TO END OF LINE

### ESC b

The Erase to End of Line function may be transmitted from the computer to the display; this causes the erasure of the display from the current cursor position to the end of the line. Erase to End of Line is not transmitted by the UNISCOPE 100 terminal to the computer.

ERASE TO END OF DISPLAY

ESC a

Erase to End of Display causes erasure of the UNISCOPE 100 terminal display from the current cursor position to the end of the display. The Erase to End of Display function is not transmitted from the UNISCOPE 100 terminal to the computer.

DELETE IN LINE

ESC c

(NOTE: DELETE, as used in this group of control functions, is equivalent to "erase and close up".)

The Delete in Line function operates the same as the Delete in Line key on the UNISCOPE 100 terminal keyboard. The character at the cursor position is deleted, and the rest of the characters in the line are moved to the left to close the gap, leaving a blank at the end of that line. Delete in Line is not transmitted from the UNISCOPE 100 terminal to the computer.

DELETE IN DISPLAY

ESC C

The Delete in Display function operates the same as the Delete in Display key (upper case) on the UNISCOPE 100 terminal keyboard. The character under the current cursor position is deleted, and all of the following characters to the end of the display are moved left one position, leaving a blank at the end of the display.

INSERT IN DISPLAY

ESC D

The Insert in Display function operates the same as the Insert in Display (lower case) key on the UNISCOPE 100 terminal keyboard. All of the characters beginning with the character under the cursor are moved to the right one position with the last character on the display being lost. A space is inserted at the cursor position. The Insert in Display character is not transmitted from the UNISCOPE 100 terminal to the computer.

INSERT IN LINE

ESC d

The Insert in Line function operates the same as the Insert in Line (lower case) key on the UNISCOPE 100 terminal keyboard. The character under the cursor and all of the characters to the right of the cursor on the same line are moved to the right with the last character on the line being lost. A space is inserted at the cursor position. The Insert in Line function is not transmitted from the UNISCOPE 100 terminal to the computer.

SCAN LEFT

ESC g

This function causes the cursor to move one position to the left. This function is not transmitted from the UNISCOPE 100 terminal to the computer.

SCAN RIGHT

ESC h

This function causes the cursor to move one space to the right, without disturbing the contents of memory.

SCAN DOWN

ESC i

This function causes the cursor to move down one line. This function is not sent from the UNISCOPE 100 terminal to the computer.

SCAN UP

ESC f

This function causes the cursor to move up one line. This function is not sent from the UNISCOPE 100 terminal to the computer.

SPACE

SP

The Space character is a nondisplay character used as a space. The space character may be transmitted from the UNISCOPE 100 terminal to the computer.

TAB

HT

The function of Tab is similar to that of the Tab key on the UNISCOPE 100 terminal keyboard. The Tab function causes the cursor to move to the next tab stop set in the memory of UNISCOPE 100 terminal.

CURSOR TO HOME

ESC e

The Cursor to Home function moves the cursor to the upper left hand corner of the screen (line 1, position 1).

INSERT LINE

ESC j

The Insert Line function causes the line at the cursor position, and all following lines, to move down one line, resulting in a blank line at the cursor position and the loss of the bottom line. The Insert Line function is not sent from the UNISCOPE 100 terminal to computer.

DELETE LINE

ESC k

The Delete Line function causes all of the lines below the line at the cursor position to move up one line, resulting in the loss of the data on the line at the cursor position and the appearance of a blank line at the bottom of the screen. Repeated use of the Delete Line function results in rolling the data upward across the screen. The Delete Line function is not transmitted from the UNISCOPE 100 terminal to the computer.

MESSAGE WAITING

BEL

The Message Waiting function is sent from the computer to the UNISCOPE 100 display to indicate that the computer has an unsolicited message available for the operator. The function lights the Message Waiting light. The actual meaning of this message and the light may vary from system to system according to local requirements. In many systems, it is used to permit the computer to present a message to the operator without destroying information being typed into the screen by the operator. BEL has this meaning only when it is included in the message before STX.

If BEL is included in the text of the message, it does not perform this function, but it is placed in memory and is not displayed.

REQUEST COMPUTER MESSAGE

BEL

When the operator presses the Message Waiting button, a message is generated for transmission to the computer; it contains the BEL character, indicating to the computer that the Message Waiting key was pressed. BEL has this meaning only if it comes before the STX message. BEL may appear in the text portion of the message as a result of having been entered either from the computer or from the auxiliary interface.

TAB STOP SET

ESC HT

The Tab Stop Set function from the computer causes the display to place the HT character in memory to act as a tab stop (see Tab). If text containing tab stops is returned to the computer, the tab stops (HT) will appear in the data. There is no restriction on the number of times HT may be used.

START BLINK FIELD

FS

Start Blink Field, FS, is placed in memory; it causes the Start Blink character to appear on the screen and blink.

END BLINK FIELD

GS

End Blink Field, GS, is placed in memory; it causes the End Blink character to appear on the screen and blink.

LOCK KEYBOARD

DC 4

When either the Transmit key or the Message Waiting key is pressed, the keyboard becomes locked (disabled). The keyboard is also locked when the Start of Text character (STX) is received from the computer. If the text message does not include the locked keyboard (DC 4) characters at the end of the Text (ETX followed by a satisfactory BCC) the keyboard is automatically unlocked. This gives the programmer the ability to lock the keyboard, and makes it unnecessary to use an unlocking character in each text message.

## PRINT (OUTPUT)

## DC 2

The Print code causes the terminal to output data from the display memory to the auxiliary channel. The format for outputting data to the auxiliary channel is identical to that for outputting data to the communications channel; that is, the area to be transmitted is to be defined by the SOE (▷) character and the cursor with suppression of nonsignificant spaces and the automatic insertion of carriage returns. The printed output thereby appears in the screen format. This is the same print function exercised by the print key (FR 4).

## PRINT TRANSPARENT (OUTPUT)

The Print Transparent function causes output from the display memory to the auxiliary channel in the same manner as the print function; neither function starts until the receipt of a satisfactory BCC associated with the message. The area transmitted is the same as in the print function, but the carriage return characters normally inserted by the logic of the UNISCOPE 100 terminal are not transmitted to the auxiliary channel. This permits program use of other functions, such as LF, for carriage return, transport function, etc. This makes the line length and card length of the device on the auxiliary channel independent of the line length of the particular display in use.

## TRANSMIT

## DC 1

The Transmit code may be transmitted from the computer to the display. It will not be acted upon until the receipt of a satisfactory BCC character in the associated message. The result of action is the same as for the depression of the transmit key (FR 6).

## START OF ENTRY (SOE) ▷

## RS

The Start of Entry character ▷ defines the beginning of the area to be transmitted to the computer or to the auxiliary interface. More than one SOE ▷ character may be on the screen (in memory at one time). The SOE ▷ closest to the cursor defines the beginning of the area that will actually be transmitted. The ability to have multiple SOE ▷ characters on the screen eliminates, for the programmer, the problem of removing these characters in instances where successive transmissions remain on the screen, and also enables the programmer to place multiple SOE ▷ characters on the screen from the computer in order to aid the formatting of the next text to be transmitted. This might be a multiple choice by the operator, where the operator positioning the cursor takes some action including a transmit which would send back a controlled area. It might be used in a situation where the program sends repeated questions or answers, with a response requested from the operator by including the SOE ▷ character at the end of the message transmitted by the program. The operator can type his response (question or answer as appropriate) beginning at the position of the cursor, and then press the transmit key at the end of his response without giving his attentions to the SOE ▷ character or to its transmitted location. This is useful when the operator is not well-trained, and where program control of the area to be transmitted is intended.

## SI - SO

The ANSCII Shift In (SI) and Shift Out (SO) characters are used to switch to an alternate graphic set, and to return to the standard. By their use, transmission is not limited to the 96 characters defined in the ANSCII character set. These characters are reserved for this function.

## LF, FF, EM

The ANSCII Line Feed (LF), Form Feed (FF), and the End of Medium (EM) are intended for use by the UNISCOPE 100 terminal for the auxiliary interface. The UNISCOPE 100 terminal itself is transparent to these characters; they may be placed in memory by either the communications interface or the auxiliary interface, and do not affect the display on the UNISCOPE 100 terminal.

## 6. MESSAGE FORMAT

A message is a sequence of characters arranged for the purpose of conveying information from an originator to one or more destinations (or addresses). There are Heading Messages that convey supervisory information and Composite Messages that convey both heading information and text.

HEADING MESSAGE	SOH ADDRESS SUPERVISORY SEQUENCE (IF REQUIRED) ETX BCC
COMPOSITE MESSAGE	SOH ADDRESS SUPERVISORY SEQUENCE (IF REQUIRED) STX TEXT ETX BCC

### 6.1. HEADING MESSAGE

A Heading Message is a sequence of characters that conveys addressing, routing, communication control and device control functions. A Heading Message is preceded by a SOH character and is ended by either an STX character or an ETX character. The terminal logic automatically generates the required header for each message. The program generates the header of each message transmitted by the computer.

### 6.2. COMPOSITE MESSAGE

The first part of a composite message is a sequence of characters forming a heading message as defined in paragraph 6.1, and terminated by STX. The second part is the text message, which is a sequence of characters conveying the text and information necessary to handle the text. The text message begins with the same STX character that terminates the heading and ends with the ETX character.

APPENDIX A. LINE CONTROL RULES FOR ANY SINGLE  
STATION OR MULTIPLEXER

Certain rules must be followed to allow recovery from telephone line error without loss or duplication of messages. These rules establish the sequence in which the various transmissions are exchanged. The rules apply to any one single station or to any one multiplexer and its associated stations. Figure A-1 illustrates the traffic flow between the UNISCOPE 100 terminal and the computer.

1. Acknowledgements from the computer to the terminal are always sent with the poll. The normal response to a message is a Poll with Acknowledge.
2. The terminal must get an acknowledgement on its next poll (specific or general) if an acknowledgement is due. If one is not received, the terminal sends the Reply Request.
3. When a "No Traffic Response" (EOT ETX) is the response to a Poll or a Poll with Acknowledge, the next poll will not include an acknowledgement because no message was received.
4. When the response of a terminal to a poll is Acknowledge without Traffic, the computer must send an Acknowledge with the next poll to prevent the terminal from sending a Reply Request.
5. After receiving a message (text) from a terminal, a Poll with Acknowledge must be sent to that station before a text message can be sent to that same station.
6. After receiving a message (TEXT) from the computer, the acknowledgement to that message must be sent with the next message to the computer from the terminal, or from any terminal recognizing the same RID.
7. After the computer receives an acknowledgement from a terminal, it must poll that terminal again before it can send another text message to that terminal.
8. A terminal does not reply with a message to a Poll with Acknowledge. The terminal will respond with a No Traffic Poll, generally EOT EOT, but for CTMC systems EOT EOT ETX BCC. The significance of this is that a single terminal never sends in two successive text messages; this prevents loss or duplication of blocks if there is an error condition. If the terminal sends a Reply Request and the computer has not received, and therefore not acknowledged the message, the computer sends a Retransmit Request to the terminal.
9. If the terminal sends in a Reply Request and the computer had already received and acknowledged the message, the acknowledgement is repeated with a poll specifically addressed to that terminal.
10. The computer treats any error in a message as a No Response and calls for a retransmission. The terminal never responds to a message containing an error. Thus, if No Response follows a Poll, the computer must repeat the Poll (without the Acknowledge if one was present). If No Response follows a Retransmit Request, the computer must repeat the Retransmit Request. If a terminal does not receive an answer to a Reply Request, the terminal must repeat the Reply Request when polled again.
11. The receipt of a message which causes a Busy condition at a terminal will be acknowledged with a WABT (DLE?), in response to the poll that follows the message which caused the busy condition. The response to subsequent specific polls will be a DLE? as long as the busy condition persists. The No Traffic response will be sent in response to subsequent general polls. The end of a Busy condition is indicated by transmission of DLE;. The DLE; response must not contain text but may contain an Acknowledge or Busy from another terminal on the multiplexer.

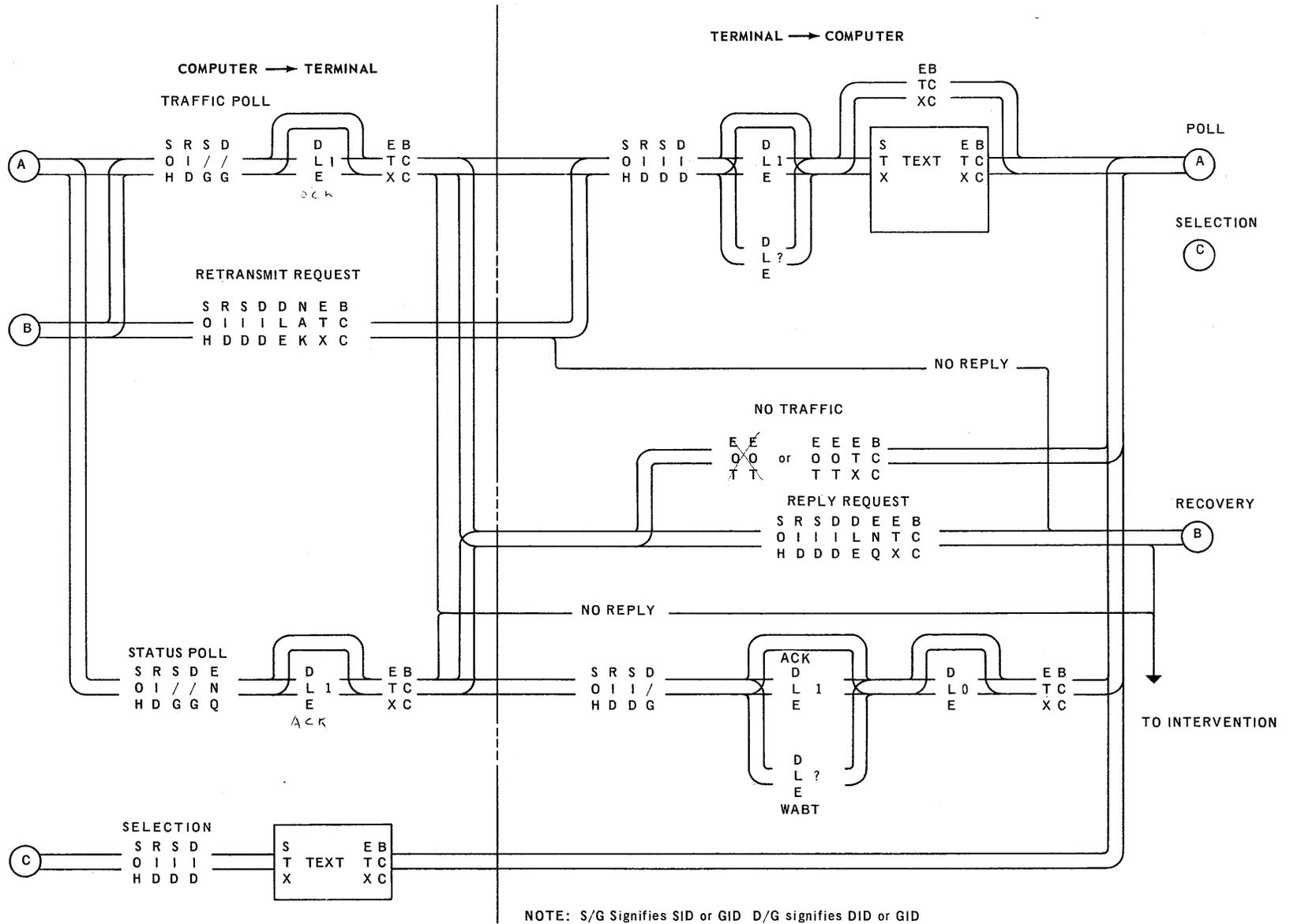


Figure A-1. UNISCOPE 100 Terminal Traffic Flow Diagram

SPERRY RAND

*all*  
*copy to*  
*ST*

**UNIVAC** DIVISION  
INTERCOMMUNICATION

Regional Field Directors  
TO: Branch Marketing Managers  
Systems Analyst Managers  
C & T Regional Sales Managers  
C & T Communications Analysts

FROM (NAME & EXT): S. C. Mehta  
LOCATION & DATE: Chicago - 16 March 1971  
DEPARTMENT & M.S.: Field Operations West

CARBONS: T. F. Lamb

SUBJECT: CMX MODEL FOR NETWORKS WITH  
UNISCOPE 100's and DCT's

A model for communications network with Uniscope 100's and DCT's was built in GPSS.

A copy of the line control rules modeled for single station and multiplexer is attached herewith. See Appendix A. The model assumes that the number of errors will be very small and ignores retransmissions and reply requests.

In one typical case for which the model was used, the following calculations were made to validate the results of the model with their help.

1. Calculations of the line busy time for:
  - a) Complete processing of each input message
  - b) Polls
  - c) Unsolicited messages
2. Calculations for line utilizations.

Central Site time calculations were made as the Central Site time is used for input to the model.

The description of modeling of polling, line control procedures and transmission of unsolicited messages is given in Appendix B.

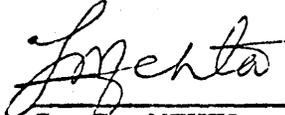
Some of the output given by the model is listed below:

- 1) Utilization of lines.
- 2) Utilization of devices for print.
- 3) Distribution of response time of all lines.
- 4) A separate distribution of response time for each of the lines.
- 5) Distribution of wait time for a locked keyboard.
- 6) Distribution of wait time for receiving a poll.
- 7) Number of polls skipped and not skipped per line and device.

Some of the questions the model can answer are:

- 1) At what load the network is saturated?
- 2) Are more lines necessary?
- 3) Is it necessary to have duplex lines?
- 4) What is the ideal interval between polls?
- 5) Does the network meet the criterion for the response time?
- 6) Should the network design be changed for improved response time?

For questions, comments, and details, feel free to contact me.

  
S. E. MEHTA

/sp

APPENDIX A. LINE CONTROL RULES FOR ANY SINGLE  
STATION OR MULTIPLEXER

Certain rules must be followed to allow recovery from telephone line error without loss or duplication of messages. These rules establish the sequence in which the various transmissions are exchanged. The rules apply to any one single station or to any one multiplexer and its associated stations. Figure A-1 illustrates the traffic flow between the UNISCOPE 100 terminal and the computer.

1. Acknowledgements from the computer to the terminal are always sent with the poll. The normal response to a message is a Poll with Acknowledge.
2. The terminal must get an acknowledgement on its next poll (specific or general) if an acknowledgement is due. If one is not received, the terminal sends the Reply Request.
3. When a "No Traffic Response" (EOT ETX) is the response to a Poll or a Poll with Acknowledge, the next poll will not include an acknowledgement because no message was received.
4. When the response of a terminal to a poll is Acknowledge without Traffic, the computer must send an Acknowledge with the next poll to prevent the terminal from sending a Reply Request.
5. After receiving a message (text) from a terminal, a Poll with Acknowledge must be sent to that station before a text message can be sent to that same station.
6. After receiving a message (TEXT) from the computer, the acknowledgement to that message must be sent with the next message to the computer from the terminal, or from any terminal recognizing the same RID.
7. After the computer receives an acknowledgement from a terminal, it must poll that terminal again before it can send another text message to that terminal.
8. A terminal does not reply with a message to a Poll with Acknowledge. The terminal will respond with a No Traffic Poll, generally EOT EOT, but for CTMC systems EOT EOT ETX BCC. The significance of this is that a single terminal never sends in two successive text messages; this prevents loss or duplication of blocks if there is an error condition. If the terminal sends a Reply Request and the computer has not received, and therefore not acknowledged the message, the computer sends a Retransmit Request to the terminal.
9. If the terminal sends in a Reply Request and the computer had already received and acknowledged the message, the acknowledgement is repeated with a poll specifically addressed to that terminal.
- \* 10. The computer treats any error in a message as a No Response and calls for a retransmission. The terminal never responds to a message containing an error. Thus, if No Response follows a Poll, the computer must repeat the Poll (without the Acknowledge if one was present). If No Response follows a Retransmit Request, the computer must repeat the Retransmit Request. If a terminal does not receive an answer to a Reply Request, the terminal must repeat the Reply Request when polled again.
11. The receipt of a message which causes a Busy condition at a terminal will be acknowledged with a WAST (DLE?), in response to the poll that follows the message which caused the busy condition. The response to subsequent specific polls will be a DLE? as long as the busy condition persists. The No Traffic response will be sent in response to subsequent general polls. The end of a Busy condition is indicated by transmission of DLE;. The DLE; response must not contain text but may contain an Acknowledge or Busy from another terminal on the multiplexer.

## APPENDIX B

This network model is very useful for marketing and response time and analysis of networks with Uniscope 100's and DCT's. It is flexible. It can be used to model any network with Uniscope 100's and DCT's by making a few changes. As long as the other networks have upto 12 communication lines and upto 90 devices, this model can be re-used by inputting the new data. Please contact me for more details about where this model can be used repetitively. The model is described below:

### POLLING:

1. The lines are polled in round-robin manner.
2. General polls are sent.
3. If the device is busy, receive wait-a-bit message.
4. Only 1 poll can be outstanding on a line.
5. The interval between polls is variable.  
Try to poll all lines at each poll interval.
6. Once a poll is received by an input, the keyboard is locked.

### LINE CONTROL PROCEDURES MODELED ARE:

- (1) INPUT: The inputs from a device come is as result of successful polls.  
(NOTE) In a modeled network, 123 characters of header information is inputted first.
- 1a. CENTRAL SITE DELAY: Delay for processing the header input.  
(NOTE) For the modeled case, see central site calculations.
- (2) OUTPUT: Send an ACKPOLL to the MUX. If no MUX on the line, send it to the device.  
(NOTE) For the modeled case, 12 characters are transmitted for an ACKPOLL.
- 2a. INPUT: An input or a NO TRAFFIC RESPONSE from the second device on the MUX.
- (3) OUTPUT: Display header to the first device. In the modeled case, 280 characters are output.
- (4) OUTPUT: Send a specific ACKPOLL if an input message was received for the ACKPOLL in Step 2, otherwise a general ACKPOLL.  
(NOTE) In the modeled case, there are 12 characters per specific ACKPOLL.

- 4a. INPUT: Receive an ACK from the first device with information about the cursor position.  
(NOTE) In the modeled case, 12 characters for ACK plus 5 for cursor position are inputted assuming a non-home cursor position.
- (5) OUTPUT: Send ACKPOLL for the above input.
- (6) OUTPUT: Display format.  
(NOTE) In the modeled case, 160 characters are sent.
- 6a. OPERATOR KEY-IN DELAY: Delay for the operator to key-in the line items.  
(NOTE) In the modeled case, this delay is of 18 seconds.
- (7) OUTPUT: Same as Step 4.
- 7a. INPUT: Receive line items from the device.  
(NOTE) In the modeled case, an average of 6 line items are received with an average of 123 characters. The maximum number of input characters reaches 280.
- If the input device is a DCT, the input is received with 160 character buffers per transmission and for each input transmission ACKPOLL is sent. See Step 2 and 2a above. Transmission of DCT output is also done in buffers of 160 characters.
- 7b. CENTRAL SITE DELAY: Delay for processing the line items.  
(NOTE) For the modeled case, see the central site time calculations.
- (8) INPUT: Receive an input message or a NTR.
- (9) OUTPUT: Display line items.  
(NOTE) In the modeled case, an average of 518 characters are sent, the maximum being 1300.
- (10) OUTPUT: Same as Step 4.
- 10a. INPUT: Same as Step 4a.
- (11) OUTPUT: Send an ACKPOLL.

(12) OUTPUT: Send the hardcopy output for printing. The device remains busy for printing.

(NOTE) In the modeled case, an average of 970 characters are sent out. The device prints at 30 characters per second. The delays for line feed and carriage return are also accounted.

(13) At the end printing of characters, the device is free for more inputs.

UNSOLICITED MESSAGES:

- (1) OUTPUT: Send a status poll to the device.
- (2) INPUT: If the device is busy, get a wait-a-bit message otherwise a NO TRAFFIC message. If the device is found to be busy, wait for it to become free.
- (3) OUTPUT: Send the output characters.  
(NOTE) In the modeled case, an average of 970 characters are sent. If they are destined for a DCT, they are transmitted as 160 characters buffers. All of these are for hardcopy printing. Hence, printing delay is also introduced.
- (4) At the end of the printing, the device is free for more inputs.

## CENTRAL SITE TIME CALCULATIONS

(The no. of instructions per file access and the wait times in these are my assumptions. For more accuracy, the C.S. model must be run.)

1.	After receiving an input of header, read 3 files.		
	2 accesses per read to FH-1782		
	For indices (3*2)*(20 mils per access.)		120
	1 access per read to 8414 disc		
	For record (3*1)*80		240
	Add wait time for 1782	6*5	30
	Add wait time for 8414	3*15	45
	Add CPU time		
	900 for translation of data etc.		9
	9 accesses*1000 for 9 accesses		9
	CPU wait time		7
	CPU speed is taken as 1 microsec/instrn		460
2.	After receiving body of the input of line items.		
	Ave. no. of line items = 6		
	Read 2 files per line items.		
	2 accesses per file read to FH-1782		
	12 reads*2*(20 mils per access)		480
	1 access per file read to 8414 disc		
	12*1*80		960
	Add wait time for FH-1782	24*5	120
	Add wait time for 8414	12*15	180
	Read 3 files per input message		
	Regardless of no. of line items.		
	3*(2 accesses/file read)*(20 mils/access of 1782)		120
	3*(1 access/file read)*(80 mils/access of 8414)		240
	1 read of 8414 to load user I plus D bank		80
	Wait time of 1782	6*5	30
	Wait time of 8414	(3plus1)*15	60
	CPU time		
	43 accesses*1000 for 43 accesses		43
	User program instructions = 5000		5
	CPU wait time		13
	Since some files will be accessed by several transactions, add some wait time for files.		169
			2500

AVERAGE NUMBER OF INPUT CHARACTERS FOR  
EVERY INPUT MESSAGE:

1.     123  
       10

2.     17

3.     123  
       10

4.     17

---

300

AVE. NO. OF OUTPUT CHARS PER TRANSMISSION  
FOR EVERY INPUT MESSAGE

1.	12	
2.	280	{ For this, Do not ignore the fact that DCT's buff can hold 160 chars.
3.	12	
4.	12	
5.	160	
6.	12	
7.	12	
8.	576	{ 518 ave. output plus 2 ACKPOLLS * 12 plus 2 * inputs of 17 chars. = 576
9.	12	
10.	12	DCTS' receive 160 char. buffs.
11.	970	{ Assume 3 xmissions of output text and 3 for ACKPOLL & 3 for ACKS from devices to ACKPOLLS.
	<hr/>	{ 970 plus 3*12 = 1006 of output 3*17 = 51 of input
	2070	

CALCULATING LINE BUSY TIME FOR  
 CONVERSATION BETWEEN CPU AND A  
 TERMINAL FOR EACH INPUT MESSAGE

MIN. NO. OF XMISSIONS = 27

FIGURING FOR 1800 B.P.S. LINES

		milliseconds
PROPAGATION TIME	8*27	216
MIN. PROPAGATION DELAYS (4 times no. prop. delay)	150*23	3450
LINE XMISSION TIME TO XMIT 2370 CHARACTERS ON AN 1800 B.P.S. LINE		10532
LINE TIME TO XMIT BACK & FORTH		<u>14198</u>
	SAY	14.2 sec.s

FOR 2400 B.P.S. LINE

XMISSION TIME		7900
PROPAGATION	8x27	216
PROP. DELAYS	150x23	3450
		<u>11566</u>
	SAY	11.6 sec.s

CALCULATING MIN. LINE BUSY TIME PER POLL AND PER NTR.

NO. OF POLL CHARS = 9

FIGURING FOR 1800 B.P.S. LINES

PROPAGATION TIME

millisecond  
8

PROPAGATION DELAY

150

LINE XMISSION TIME FOR 9 CHARS. AT 10 BITS/CHAR

50  
40

198  
208

FOR 2400 B.P.S. LINE

PROPAGATION TIME

8

PROPAGATION DELAY

150

LINE XMISSION TIME FOR 9 CHARS.

30

188

NTR

FOR 1800 B.P.S. LINES

PROPAGATION TIME

8

PROPAGATION DELAY

150

LINE XMISSION TIME FOR 5 CHARS.

28  
22

180  
186

FOR 2400 B.P.S. LINES

PROPAGATION TIME

8

PROPAGATION DELAY

150

LINE XMISSION TIME FOR 5 CHARS.

17

175

CALCULATING MIN. NO. OF CHARS. FOR EVERY UNSOLICITED (OUTPUT) MESSAGE

	no. of chars.
AVE. NO. OF OUTPUT CHARS. (USM ONLY)	970
{ DCT 1000 GET BUFF.S OF 160 CHARS.	
{ FIGURE ABOUT 4 ACKPOLLS FOR EACH USM.	
4*12 CHARS PER ACKPOLL	48
MOST USMS GO TO DCTS.	
4 ACKS FROM TERMINALS FOR EACH USM.	
4*17 CHARS per ACK	68
1 STATUS POLL * 9 CHARS PER POLL	9
1 REPLY TO STATUS POLL * 5 CHARS/REPLY	5
	<u>1100</u>

NOTE: NO. OF XMISSIONS = 14

CALCULATING MIN. LINE BUSY TIME FOR EVERY UNSOLICITED (OUTPUT) MESSAGE

FIGURING FOR 1800 B.P.S. LINES.

milliseconds

THERE ARE 14 XMISSIONS/USM

PROPAGATION TIME 14\*8 112

PROPAGATION DELAY 14\*150 2100

LINE XMISSION TIME FOR 1100 CHARS 4778

6990

SAY 7.0 sec.

FIGURING FOR 2400 B.P.S. LINES.

PROPAGATION TIME 14\*8 112

PROPAGATION DELAY 14\*20 2100

LINE XMISSION TIME FOR 1100 CHARS. 3667

5879

SAY 5.9 sec.

Table 3-6. SID Address Codes (Cont)

SID Code (Octal)	Connector Part No.	Bend Pins Up
153	2807774-60	1-11
154	2807774-61	1-3-13
155	2807774-62	1-13
156	2807774-63	1-3
157	2807774-64	1

Table 3-7. I/O-2 Board (A12) Strapping Procedure

Step	Procedure / Function	Figure Reference
1	Select Interface Mode (Synchronous or Asynchronous) following table 3-8	3-4
2	Check system operation (Univac or IEM)	3-4
3	Select Baud Rate following table 3-9	3-4

Table 3-8. Interface Mode Selection (A12)

System Operation	Connector Location on A12	Synchronous		Asynchronous	
		Connector Part Number	Bend Pins Up	Connector Part Number	Bend Pins Up
RS232C/V.24	Z24	2807774-00	None	2807774-00	None
	Z48	2807774-40	1-7-13	2807774-65	3-7-13
	Z60	2807774-00	None	2807774-64	1
	Z72	2807774-14	5-7-9	2807774-15	7-9-11
MIL-STD-188C (See Note)	Z36	2807774-69	4-7	2807774-00	None
	Z48	2807774-09	3-5-7-9-11	2807774-66	1-5-7-9-11
	Z60	2807774-70	2-3-4-6-7-8	Not used	empty socket
	Z72	2807774-14	5-7-9	2807774-15	7-9-11
Multiplexer	Z48	2807774-09	3-5-7-9-11	2807774-66	1-5-7-9-11
	Z60	2807774-00	None	2807774-00	None
	Z72	2807774-33	1-3-5-7-11-13	2807774-33	1-3-5-7-11-13
	Z84	2807774-00	None	2807774-00	None

NOTE

MIL-STD-188C specifies a noninverted clock signal for synchronous strapping. MIL-STD-188B specifies either an inverted or noninverted clock signal, and SPERRY UNIVAC modems contain provisions to allow either configuration. Determine modem configuration and change the modem, if necessary, to conform to MIL-STD-188C (noninverted) clock signal.

Table 3-9. Baud Rate Selection (A12)

Asynchronous		
Baud Rate (bps)	Connector Part Number at Location XZ133	Bend Pins Up
300	2807774-67	1-3-9-11-13
600	2807774-34	1-3-5-7-13
1200	2807774-19	3-9-11-13
1600	2807774-42	1-3-5-7
1800	2807774-38	1-5-7-13
2400	2807774-68	3-5-7-13
Synchronous		
Baud Rate (bps)	Connector Part Number at Location XZ84	Bend Pins Up
2400	2807774-62	1-13
4800	2805281-17	3-13
9600	2807774-63	1-3

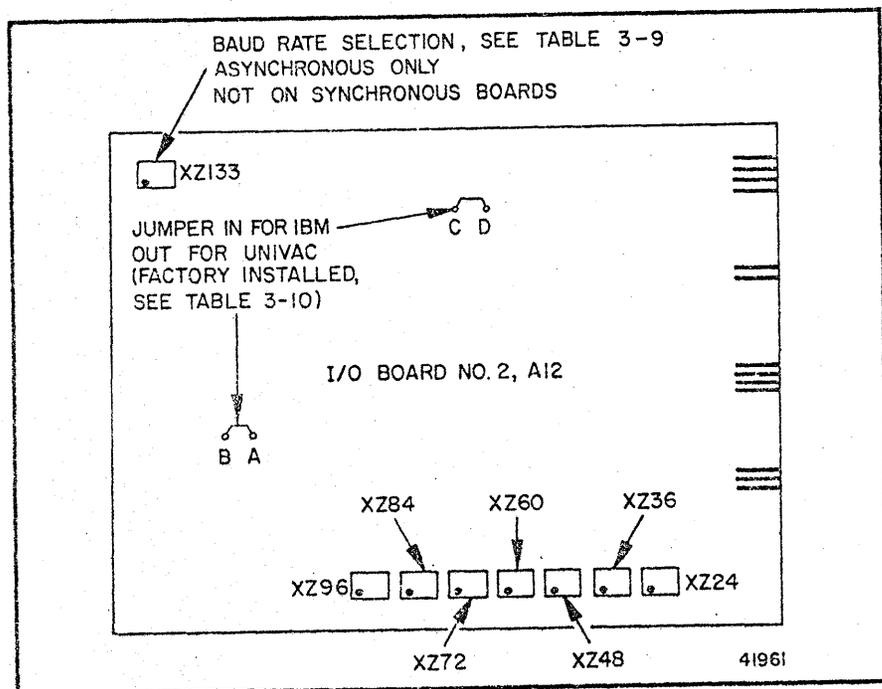


Figure 3-4. I/O-2 Board (A12) Strapping