EK-VT220-RM-001

VT220

Programmer Reference Manual

Digital Equipment Corporation

1st Edition, August 1983

The information in this document is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

CONTENTS

CHAPTER 4 RECEIVED CODES

TABLES

FIGURES

PREFACE

This reference manual is provided for use by people with knowledge of basic computer programming. It provides the information needed to access the VT22Ø features. The manual is organized into the following chapters and appendices:

Chapter 1 Terminal Overview provides an introduction to the terminal. It discusses breifly the terminal's capabilities and operating modes.

Chapter 2 Character Encoding describes the character encoding schemes used for the terminal. It also describes the terminal's character sets and provides an overview of control function format.

Chapter 3 Transmitted Codes describes the codes that the terminal transmits to a program.

Chapter 4 Received Codes describes the codes that the terminal recognizes and responds to.

Appendix A VT22Ø/VT1Ø2 Differences describes the major differences between a VT1Ø2 terminal and the VT22Ø operating in a VT1ØØ mode.

Appendix B Additional VT22Ø Documents provides a list and description of additional VT22Ø documents that can be ordered from DIGITAL.

CHAPTER 1
TERMINAL OVERVIEW

1.1 GENERAL

This chapter provides an overview of the VT220 terminal. It describes the VT220's major characteristics, capabilities, and operating modes.

The VT220 (Figure 1-1) is a general-purpose, video display, terminal packaged as a "two-box" terminal monitor/system unit and keyboard that uses ANSI Standard functions.

1.2 TERMINAL CHARACTERISTICS AND CAPABILITIES

The following subparagraphs summarize major VT220 features.

1.2.1 Display Characteristics and Capabilities

A summary of the major display characteristics is as follows:

- Monochrome monitor
- 24 rows X 80/132 columns
  7 X 10 dot matrix in 10 X 10 cell for 80 columns
  7 X 10 dot matrix in 9 X 10 cell for 132 columns
- 800 (horizontal) X 240 (vertical) pixels (80 columns)
  1188 (horizontal) X 240 (vertical) pixels (132 columns)

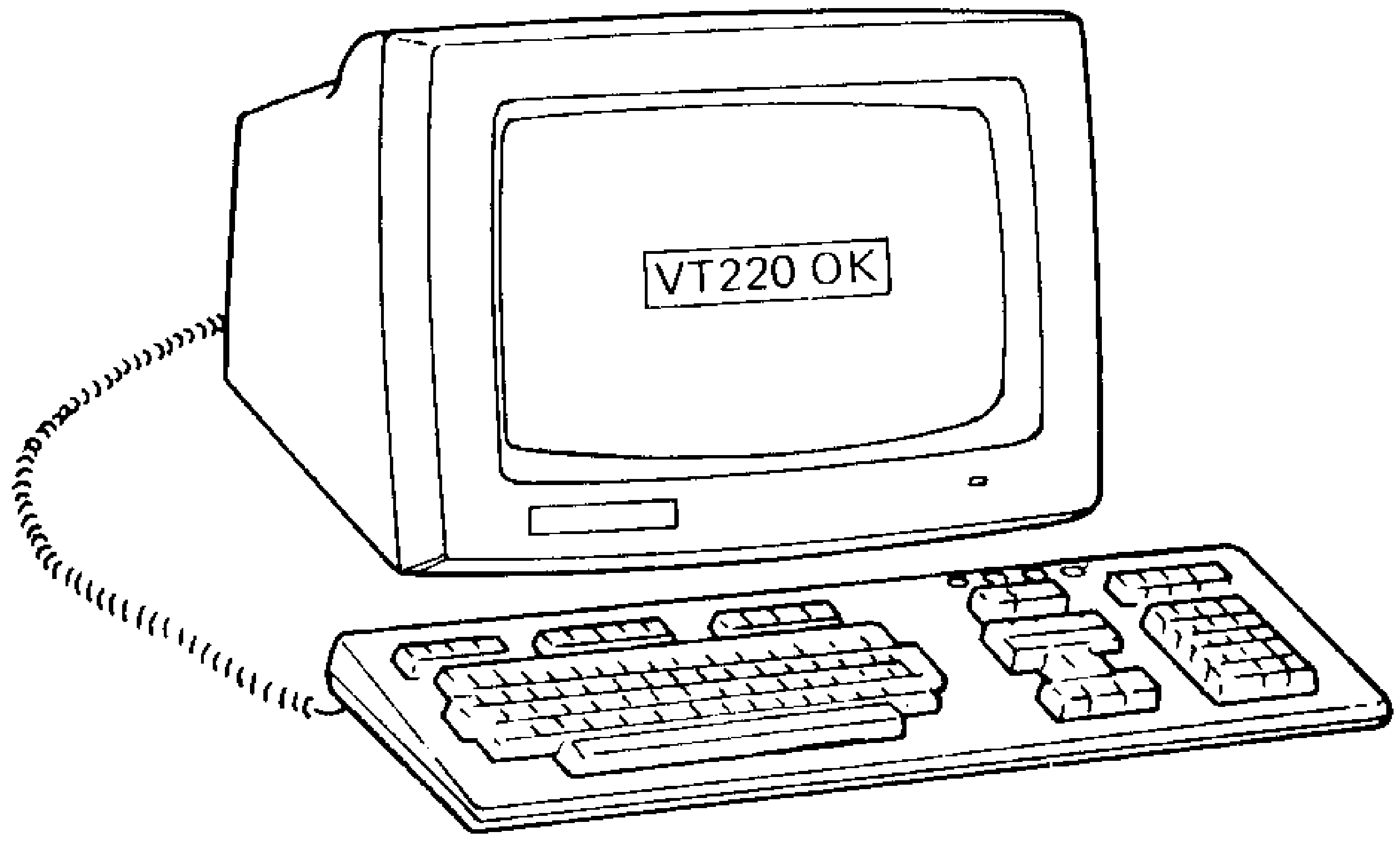


Figure 1-1 VT220 Video Display Terminal

1.2.2 Text Capabilities

A summary of the major text capabilities is as follows:

- Five character sets of 94 characters each (including DEC Multinational Character Set)
- Down-line loadable character set
- User-definable function keys
- Reverse video character attribute
- Underline character attribute
- Bold character attribute
- Blink character attribute
- Double-height/double width line attribute
- ANSI compatible control functions
- VT52 mode

1.3 COMMUNICATION ENVIRONMENT

A summary of major communications features is as follows:

- Asynchronous communications to 19.2K bits per second
- EIA RS232C host port
- 2Ø mA host port
- EIA RS232C printer port
- 7- and 8-bit character format

1.4 MAJOR OPERATING STATES

The VT22Ø has three major operating states:

- Set-Up
- On-Line
- Local

Set-up is selected from the keyboard. This operating state is used by the terminal operator to configure or examine terminal operating characteristics such as transmit and receive speeds. On-Line and Local are selected using Set-up.

On-Line (selectable only in Set-up) is used to let the terminal communicate with a host computer. When the terminal is on-line, data entered at the keyboard is transmitted to the host computer and data received from the host computer is displayed on the monitor. A local-echo feature (also selectable in Set-up) routes data entered at the keyboard to the monitor as well as to the host computer.

Local (selectable only in Set-up) is used to effectively place the host computer on "hold." When the terminal is in local, data entered at the terminal keyboard is sent only to the terminal monitor. Data received from the host computer is buffered and redirected to the terminal monitor after the terminal is returned to on-line.

1.5 MAJOR OPERATING MODES

The VT22Ø has the following major operating modes selectable from the keyboard or from the host via control codes:

- VT1ØØ Mode
- VT2ØØ Mode, 7-Bit Controls
- VT2ØØ Mode, 8-Bit Controls
- VT52 Mode

VT1ØØ Mode executes standard ANSI functions. It has a high degree of compatibility with DIGITAL's VT1Ø2 terminal (VT22Ø/VT1Ø2 differences are described in Appendix A). This mode restricts use of the keyboard to VT1ØØ keys. All data is restricted to 7 bits and only ASCII, U.K., or Special Graphics Characters are generated. This mode is provided for strict backward compatibility with existing software written for the VT1ØØ terminal family.

VT2ØØ Mode, 7-Bit Controls executes standard ANSI functions. This mode (factory default mode) provides full range of VT22Ø capabilities in an 8-bit communications environment with 7-bit controls (see Chapter 2) This mode supports the DEC Multinational Character Set and provides some backward compatibility for existing VT1ØØ software.

VT2ØØ Mode, 8-Bit Controls executes standard ANSI functions. This mode provides full range of VT22Ø capabilities in an 8-bit communications environment with 8-bit controls (see Chapter 2). This mode supports the DEC Multinational Character Set.

VT52 Mode executes DIGITAL private functions (not ANSI). It has a degree of compatibility with DIGITAL's VT1Ø2 terminal operating in a VT52 mode. This mode restricts use of the keyboard to VT52 keys. All data is restricted to 7 bits, and only ASCII, U.K. National characters, or Special Graphics characters are generated.

CHAPTER 2
CHARACTER ENCODING

2.1 GENERAL

This chapter describes the character encoding used on the VT220 when operating in text mode. It also describes the VT220 character sets and provides an overview of the control functions. You must have a basic understanding of the coding concepts presented in this chapter before using the control functions described in Chapters 3 and 4.

2.2 CODING STANDARDS

The VT220 uses an 8-bit character encoding scheme and a 7-bit code extension technique that are compatible with the following ANSI and ISO standards. ANSI (American National Standards Institute) and ISO (International Organization for Standardization) specify the current standards for character encoding used in the communications industry.

Standard	Description
ANSI X3.4 - 1977	American Standard Code for Information Interchange (ASCII)
ISO 646 -- 1977	7-Bit Coded Character Set for Information Processing Interchange
ANSI X3.41 -- 1974	Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Code Information Interchange
ISO Draft International Standard 2022.2	7-Bit and 8-Bit Coded Character Sets -- Code Extension Techniques
ANSI X3.32 -- 1973	Graphic Representation of the Control Characters of American National Code for Information Interchange
ANSI X3.64 -- 1979	Additional Controls for Use with American National Standard for Information Interchange

Standard	Description
ISO Draft International Standard 6429.2	Additional Control Functions for Character Imaging Devices

2.3 CODE TABLE

A code table is a convenient way to represent 7- and 8-bit characters, because groupings of characters and their relative codes can be seen readily.

2.3.1 7-Bit ASCII Code Table

Table 2-1 is the 7-bit ASCII code table. There are 128 positions corresponding to 128 character codes arranged in a matrix of 8 columns and 16 rows.

Each row represents a possible value of the four least significant bits of a 7-bit code (Figure 2-1). Each column represents a possible value of the three most significant bits.

Table 2-1 shows the octal, decimal, and hexadecimal code for each ASCII character. You can also represent any character by its position in the table. For example, the character "H" (column 4, row 8) can be represented as 4/8. This column/row notation is used to represent characters and codes throughout this manual. For example:

```
1/11 2/3 3/6
ESC   #   6
```

means that the ESC character is at column 1, row 11; the character # is at column 2, row 3; and the character 6 is at column 3, row 6.

The VT22Ø processes received characters based on two character types defined by ANSI: graphic characters and control characters.

Graphic characters are characters you can display on a video screen. The ASCII graphic characters are in positions 2/1 through 7/14 of Table 2-1. They include all American and English alphanumeric characters plus punctuation marks and various text symbols. Examples are: C, n, ", !, +, $ (the English pound sign is not an ASCII graphic character).

Control characters are non-displayed single-byte codes that perform specific functions in data communications and text processing. The ASCII control characters are in positions Ø/Ø through 1/15 (columns Ø and 1) of Table 2-1. The SP character ("space", 2/Ø) can be considered either a graphic character or a control character depending on the context. DEL (7/15) is always used as a control character.

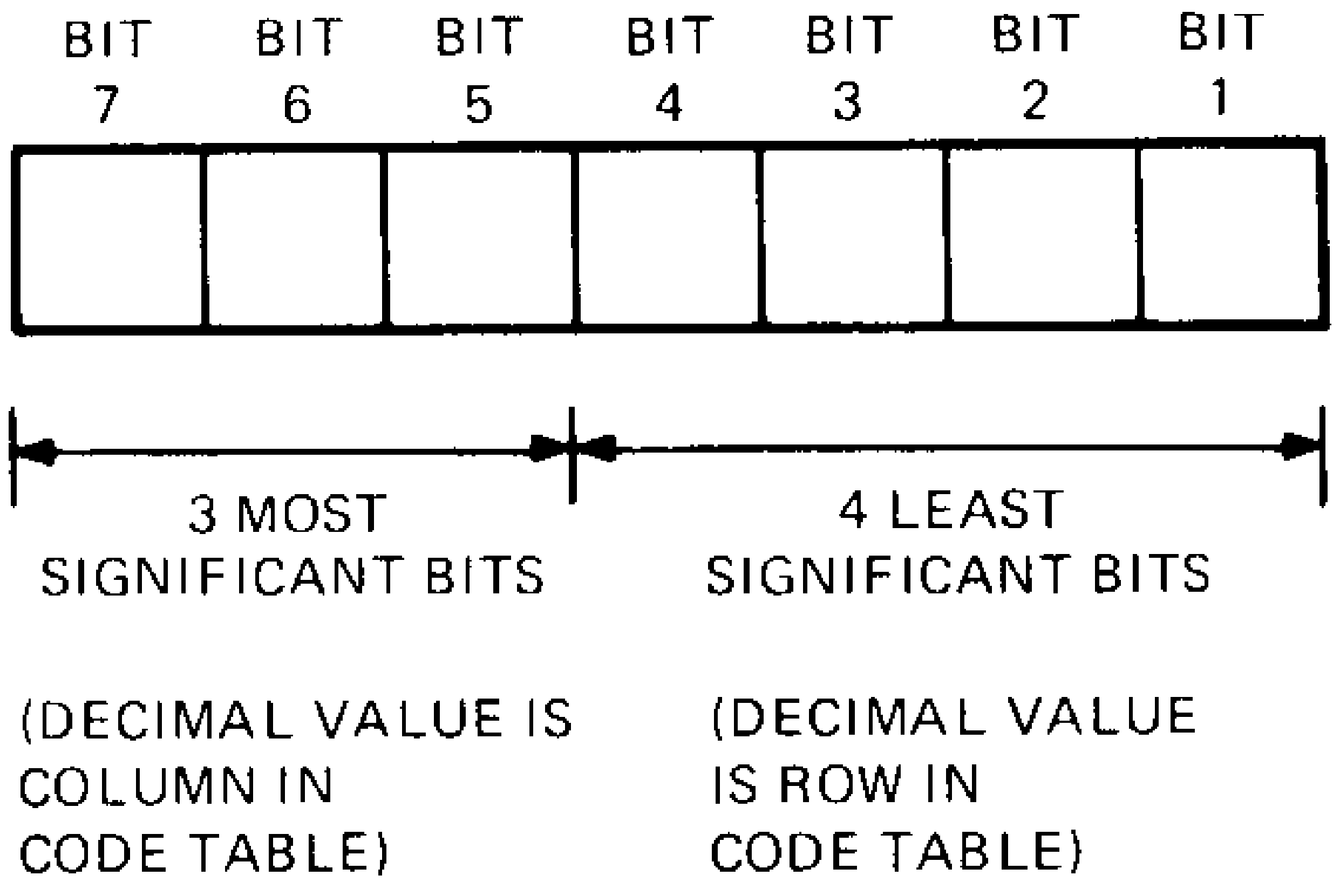
Table 2-1 7-Bit ASCII Code Table

ROW	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893A-83



MA-0890-83

Figure 2-1   7-Bit Code

The codes and function of control characters have been standardized by ANSI. Examples of ASCII control characters with their ANSI-standard mnemonics are: CR (carriage return), FF (form feed), CAN (cancel).

2.3.2 8-Bit Code Table

The above conventions can be generalized to the 8-bit character encoding used on the VT22Ø. Table 2-2 shows the 8-bit code table. It has twice as many columns as the 7-bit table, because it contains 256 versus 128 code values.

As with the 7-bit table, each row represents a possible value of the four least significant bits of an 8-bit code (Figure 2-2). Each column represents a possible value of the four most significant bits.

All codes on the left half of the 8-bit table (columns Ø through 7) are 7-bit compatible: their 8th bit is not set and can be ignored or assumed to be Ø. You can use these codes in either a 7-bit or an 8-bit environment. All codes on the right half of the table (columns 8 through 15) have their 8th bit set. You can use these codes only in an 8-bit compatible environment.

The 8-bit code table (Table 2-2) has two sets of control characters (CØ "control zero", and Cl "control one") and two sets of graphic characters (GL "graphic left", and GR "graphic right").

On the VT22Ø, the basic functions of the CØ and Cl codes are as defined by ANSI. CØ codes represent the ASCII control characters described earlier. The CØ codes are 7-bit compatible. The Cl codes represent 8-bit control characters that let you perform additional functions beyond those possible with the CØ codes. Cl codes can be used directly only in an 8-bit environment. Some Cl code positions have been left blank because their functions have not yet been standardized.

NOTE: The VT22Ø does not recognize all CØ and Cl codes. Chapter 4 identifies and describes the ones it does; all others are simply ignored (no action taken).

The GL and GR sets of codes are reserved for graphic characters. There are 94 GL codes in positions 2/1 through 7/14 and 94 GR codes in positions 1Ø/1 through 15/14. By ANSI standards, positions 1Ø/Ø and 15/15 are not used. You can use GL codes in 7-bit or 8-bit environments. You can use GR codes only in an 8-bit environment.

Table 2-2 8-Bit Code Table

COLUMN ROW	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP							DCS	///					
01	SOH	DC1								PU1						
02	STX	DC2								PU2						
03	ETX	DC3								STS						
04	EOT	DC4							IND	CCH						
05	ENQ	NAK							NEL	MW						
06	ACK	SYN							SSA	SPA						
07	BEL	ETB							ESA	EPA						
08	BS	CAN							HTS							
09	HT	EM							HTJ							
10	LF	SUB							VTS							
11	VT	ESC							PLD	CSI						
12	FF	FS							PLU	ST						
13	CR	GS							RI	OSC						
14	SO	RS							SS2	PM						
15	SI	US						DEL	SS3	APC						///

C0 CODES (columns 00–01) — GL CODES (columns 02–07) — C1 CODES (columns 08–09) — GR CODES (columns 10–15)

7-BIT CODE TABLE (columns 00–07)

MA-0892-83

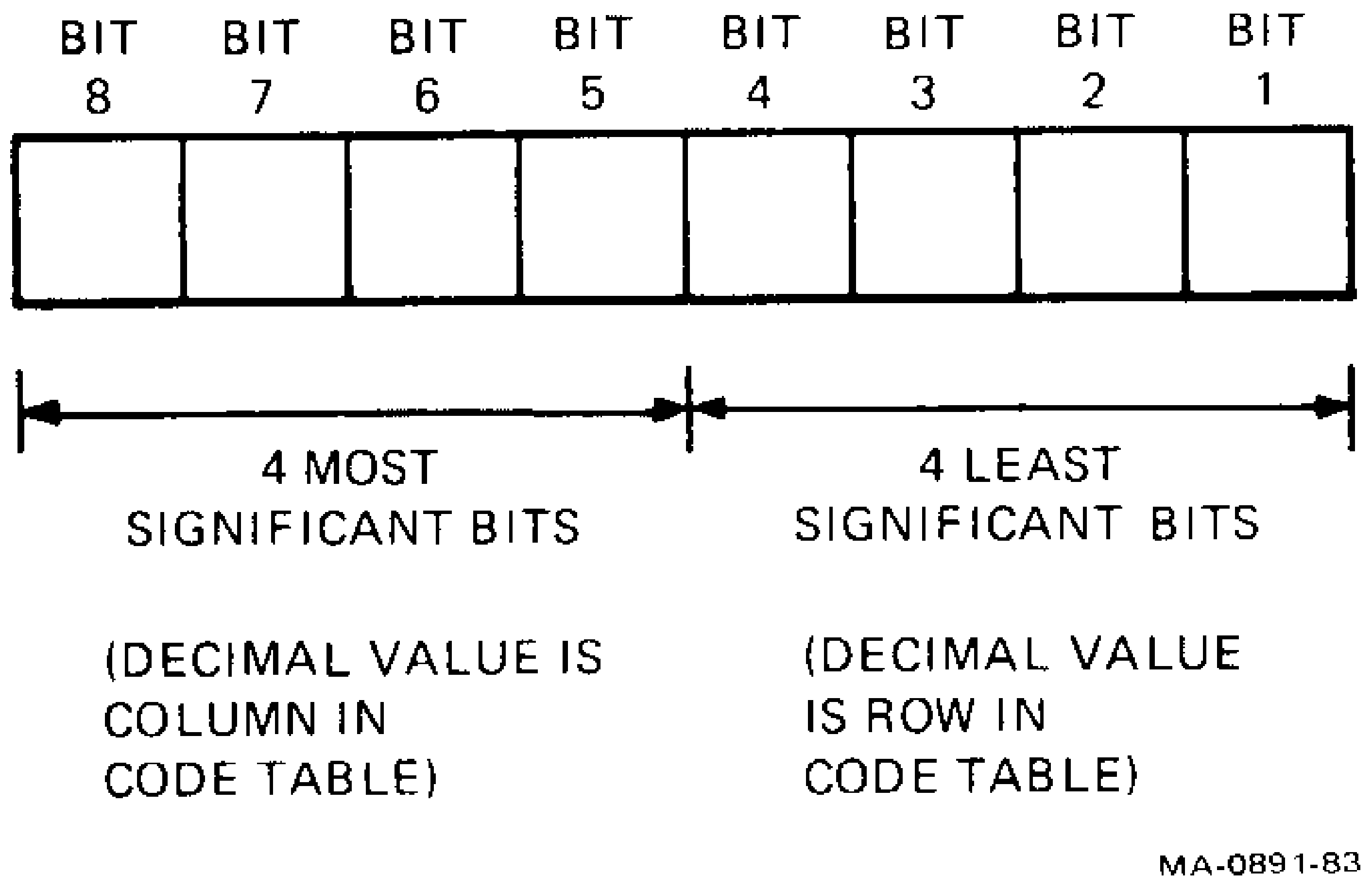


Figure 2-2   8-Bit Code

2.4 CHARACTER SETS

You cannot change the functions of the CØ or C1 codes. However, you can "map" different sets of graphic characters into the GL and/or GR codes. The sets are stored "on call" within the terminal as a graphic repertoire. But they are not available for use until mapped into the GL or GR codes. The commands for mapping graphic character sets into GL or GR are described in Chapter 4.

The terminals graphic repertoire consists of the following character sets which are described in the following subparagraphs.

- DEC Multinational Character Set (comprising ASCII Graphics Set and the DEC Supplemental Graphics Set)
- U.K. National Set
- DEC Special Graphics Set
- Down-Line Loadable Character Set

2.4.1 DEC Multinational Character Set

By factory default, when you power up or reset the terminal, the DEC Multinational Set is mapped into the 8-bit code matrix (columns Ø through 15). The DEC Multinational Character Set is shown in Table 2-3.

The 7-bit compatible left half of the DEC Multinational Set is the ASCII Graphics Set: the CØ codes are the ASCII control characters and the GL codes are the ASCII Graphics Set.

The 8-bit compatible right half of the DEC Multinational Set includes the C1 8-bit control characters in columns 8 and 9. The GR codes are the DEC Supplemental Graphics Set. The DEC Supplemental Graphic Character Set has alphabetic characters with accents and diacritical marks that appear in the major Western European alphabets. It also has other symbols not included in the ASCII Graphics Set.

The terminal accommodates over a dozen national (Western European) keyboards. All keyboards assume the default DEC Multinational Character Set mapping. In addition, the code descriptions in the rest of this manual assume this mapping. Various characters from the DEC Supplemental Graphics Set appear as standard (printing character) keys on different keyboards.

The DEC Supplemental Graphic Character Set is not available in the VT52 and VT1ØØ modes.

Table 2-3 DEC Multinational Character Set
(CØ and GL Codes) (Sheet 1 of 2)

ROW	COLUMN	0		1		2		3		4		5		6		7	
	b8 b7 b6 b5 BITS b4 b3 b2 b1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

|← C0 CODES →|← GL CODES (ASCII GRAPHICS) →|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83

Table 2-3 DEC Multinational Character Set (C1 and GR Codes)(Sheet 2 of 2)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80	DCS	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81	PU1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82	PU2	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83	STS	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84	CCH	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85	MW	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
SSA	206 134 86	SPA	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
HTJ	211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 733 E9	ù	371 249 F9	1 0 0 1	9
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
PLD	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
PLU	214 140 8C	ST	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D	OSC	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E	PM	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F	APC	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	15

|←— C1 CODES —→|←— GR CODES (DEC SUPPLEMENTAL GRAPHICS) —→|

MA 0894-83

2.4.2 U.K. National Set

The terminal's graphic repertoire includes the U.K. National Set. This set is shown in Table 2-4. This set differs from the ASCII Character Set only in that the pound sign (£) replaces the number sign (#) in column 2, row 3 (2/3). This set is available for use only in the VT1ØØ or VT52 modes.

Table 2-4 UK National Character Set

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS b8 b7 b6 b5 b4 b3 b2 b1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		18	HEX

MA-08938-83

2.4.3 DEC Special Graphics Set

The terminal's graphic repertoire includes the DEC Special Graphics Set (also known as the VT100 Line Drawing Character Set). This character set is shown in Table 2-5. It has about two thirds of the ASCII graphic characters. In addition it has special symbols and short line segments. The line segments let you create a limited range of pictures while still using text mode.

Commands described in Chapter 4 let you map the DEC Special Graphics Set into either GL or GR, replacing either the ASCII Graphics Set or the DEC Supplemental Graphics Set. The recommended mapping is to switch between ASCII and DEC Special Graphics in GL because the latter has most of the ASCII graphic characters. Also, this mapping is compatible with a VT100 terminal.

Table 2-5 Special Graphics

	COLUMN	0	1	2	3	4	5	6	7
	BITS B7 B6 B5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
ROW	B4 B3 B2 B1								
0	0 0 0 0	NUL 0 0 0	DLE 20 16 10	SP 40 32 20	0 60 48 30	@ 100 64 40	P 120 80 50	◆ 140 96 60	⎻ SCAN 3 160 112 70
1	0 0 0 1	SOH 1 1 1	DC1 (XON) 21 17 11	! 41 33 21	1 61 49 31	A 101 65 41	Q 121 81 51	▒ 141 97 61	─ SCAN 5 161 113 71
2	0 0 1 0	STX 2 2 2	DC2 22 18 12	" 42 34 22	2 62 50 32	B 102 66 42	R 122 82 52	␉ 142 98 62	⎼ SCAN 7 162 114 72
3	0 0 1 1	ETX 3 3 3	DC3 (XOFF) 23 19 13	# 43 35 23	3 63 51 33	C 103 67 43	S 123 83 53	␌ 143 99 63	⎽ SCAN 9 163 115 73
4	0 1 0 0	EOT 4 4 4	DC4 24 20 14	$ 44 36 24	4 64 52 34	D 104 68 44	T 124 84 54	␍ 144 100 64	├ 164 116 74
5	0 1 0 1	ENQ 5 5 5	NAK 25 21 15	% 45 37 25	5 65 53 35	E 105 69 45	U 125 85 55	␊ 145 101 65	┤ 165 117 75
6	0 1 1 0	ACK 6 6 6	SYN 26 22 16	& 46 38 26	6 66 54 36	F 106 70 46	V 126 86 56	° 146 102 66	┴ 166 118 76
7	0 1 1 1	BEL 7 7 7	ETB 27 23 17	' 47 39 27	7 67 55 37	G 107 71 47	W 127 87 57	± 147 103 67	┬ 167 119 77
8	1 0 0 0	BS 10 8 8	CAN 30 24 18	(50 40 28	8 70 56 38	H 110 72 48	X 130 88 58	␤ 150 104 68	│ 170 120 78
9	1 0 0 1	HT 11 9 9	EM 31 25 19	) 51 41 29	9 71 57 39	I 111 73 49	Y 131 89 59	␋ 151 105 69	≤ 171 121 79
10	1 0 1 0	LF 12 10 A	SUB 32 26 1A	* 52 42 2A	: 72 58 3A	J 112 74 4A	Z 132 90 5A	┘ 152 106 6A	≥ 172 122 7A
11	1 0 1 1	VT 13 11 B	ESC 33 27 1B	+ 53 43 2B	; 73 59 3B	K 113 75 4B	[133 91 5B	┐ 153 107 6B	π 173 123 7B
12	1 1 0 0	FF 14 12 C	FS 34 28 1C	, 54 44 2C	< 74 60 3C	L 114 76 4C	\ 134 92 5C	┌ 154 108 6C	≠ 174 124 7C
13	1 1 0 1	CR 15 13 D	GS 35 29 1D	- 55 45 2D	= 75 61 3D	M 115 77 4D	] 135 93 5D	└ 155 109 6D	£ 175 125 7D
14	1 1 1 0	SO 16 14 E	RS 36 30 1E	. 56 46 2E	> 76 62 3E	N 116 78 4E	^ 136 94 5E	┼ 156 110 6E	· 176 126 7E
15	1 1 1 1	SI 17 15 F	US 37 31 1F	/ 57 47 2F	? 77 63 3F	O 117 79 4F	(BLANK) 137 95 5F	⎺ SCAN 1 157 111 6F	DEL 177 127 7F

|←— C0 CODES —→|← GL CODES (DEC SPECIAL GRAPHICS) —→|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893C-83

2.4.4 Down-Line Loadable Character Set

The terminal provides for a 94-character down-line loadable graphic character set. You can define this character set and map it into either GL or GR as described in Chapter 4. This feature is available only in the VT2ØØ mode.

2.5 CONTROL FUNCTIONS

You use control functions in your program to specify how the terminal should handle data. There are many uses for control functions. Here are some examples:

- Move the cursor on the display
- Delete a line of text from the display
- Change character and line attributes
- Change graphic character sets
- Set the terminal operating mode

You can use all control functions in text mode and express them as single-byte or multi-byte codes.

The single-byte codes are the CØ and C1 control characters. Your program can perform a limited number of functions using CØ characters. C1 characters give you a few more functions, but your program can use them directly only in an 8-bit environment.

Multi-byte control codes represent far more functions because of the variety of code combinations possible. These codes are called escape sequences, control sequences, and device control strings. Some sequences are ANSI standardized and used throughout the industry; others are "private" sequences created by manufacturers like Digital for specific families of products. Private sequences, are with ANSI standardized sequences, obey ANSI standards governing character code composition.

2.5.1 Escape Sequences

An escape sequence is a sequence of one or more ASCII Graphic characters preceded by the CØ character ESC (1/11). For example,

```
1/11 2/3 3/6
ESC   #   6
```

is an escape sequence that causes the current line of text to have double-width characters.

Because escape sequences use only 7-bit characters, you can use them in 7-bit or 8-bit environments.

NOTE: When using escape or control sequences, remember that it is the code that defines a sequence -- not the graphic representation of the characters. The characters are shown for readability only and presume the DEC Multinational Character Set mapping (ASCII Graphics Set in GL and DEC Supplemental Graphics Set in GR).

An especially important use of escape sequences is extending the functionality of 7-bit control functions. ANSI permits you to use two-byte escape sequences as 7-bit code extensions to express each of the C1 control codes. This is a valuable feature when your application must be compatible with a 7-bit environment. For example, the C1 characters CSI, SS3, and IND can be expressed as follows:

C1 Character	7-Bit Code Extension Equivalent (escape sequence)
9/11 CSI	1/11 5/11 ESC [
8/15 SS3	1/11 4/15 ESC O
8/4 IND	1/11 4/4 ESC D

The above code extension technique can be generalized as follows: You can express any C1 control character as a two-character escape sequence whose second character has a code that is 40 (hexadecimal) and 64 (decimal) less than that of the the C1 character. Conversely, you can make any escape sequence whose second character is in the range of 4/0 through 5/15 one byte shorter by removing the ESC and adding 40 (hexadecimal) to the code of the second character, thereby generating an 8-bit control character.

2.5.2 Control Sequences

A control sequence is a sequence of one or more ASCII graphic characters preceded by CSI (9/11). But CSI (9/11) can also be expressed as the 7-bit code extension ESC [(1/11 5/11). So you can express all control sequences as escape sequences whose second character code is [(5/11). For example, the following two sequences are equivalent sequences that perform the same function (they cause the display to use 132 columns per line rather than 80).

```
9/11 3/15 3/3 6/8
CSI   ?    3    h
```

```
1/11 5/11 3/15 3/3 6/8
ESC   [    ?    3   h
```

Whenever possible, you should use CSI instead of ESC [to introduce a control sequence. CSI uses one less byte than ESC [, so you gain processing speed. But a sequence starting with CSI can be used only in an 8-bit environment (because CSI is a C1 control character).

2.5.3 Device Control Strings

A device control string is a delimited string of characters which is used in a data stream as a logical entity for control purposes. It consists of an opening delimiter (a device control string introducer), a command string (data), and a closing delimiter (a string terminator).

You use device control strings to down-line load character sets definitions for user-defined keys.

In the VT22Ø, the device control string format is:

9/Ø	....	9/12
DCS	Data	ST
Device Control String (opening delimiter)	.UDK .Character Set	String Terminator (closing delimiter)

DCS is an 8-bit control character. You can also express it as ESC P (1/11 5/Ø) when coding for a 7-bit environment.

ST is an 8-bit control character. You can also express it as ESC \ (1/11 5/12) when coding for a 7-bit environment.

2.6 WORKING WITH 7- AND 8-BIT ENVIRONMENTS

To take advantage of the terminal 8-bit character set, your program and communication environment must be 8-bit compatible and the terminal must operate in a VT2ØØ mode. When operating in the VT1ØØ or VT52 mode, you are limited to working in a 7-bit environment. The following subparagraphs describe conventions when operating in a VT2ØØ mode.

2.6.1 Conventions for Codes Transmitted to the Terminal

The terminal expects to receive character codes in a form consistent with 8-bit coding. Your application can freely use the CØ and C1 control codes as well as the 7-bit C1 code extensions if necessary. The terminal always interprets these codes properly. All code extensions you may need to use in your application are documented in Chapter 4 together with the equivalent C1 control codes.

When your program sends GL or GR codes, the terminal interprets these according to the graphic character mapping currently being used. The factory default mapping, which is set when you power up or reset the terminal is the DEC Multinational Character Set. This mapping assumes the current terminal mode is one of the VT2ØØ modes.

2.6.2 Conventions for Codes Transmitted by the Terminal

Codes transmitted by the terminal to a program come either from the keyboard or possibly in response to command issued from the host (application program or operating system). In the VT200 mode, the terminal always transmits all GL and GR graphic codes to the application exactly as they are generated regardless of whether the application handles 8-bit codes properly or not. If, however, a 7-bit communications line is used, C1 controls are transmitted as escape sequences and the terminal does not allow the generation of 8-bit graphic codes.

Most function keys on the keyboard generate multi-byte control codes. Many of these codes start with either CSI (9/11) or SS3 (8/15) which are C1 characters. If your application environment cannot handle 8-bit codes, you can make the terminal automatically convert all C1 codes to their equivalent 7-bit code extensions before sending them to the application by using DECSCL commands described in Chapter 4.

By factory default, the terminal is set to automatically convert all C1 codes transmitted to the application to 7-bit code extensions. However, to ensure the proper mode of operation, always use the appropriate DECSCL commands described in Chapter 4.

NOTE: New programs should be capable of accepting both 7 and 8-bit forms of the C1 controls.

2.7 DISPLAY CONTROLS MODE

The terminal has a "Display Controls" mode that lets you prevent execution of most of the control codes and, instead, display them as graphic characters for debugging purposes. This mode is available through use of the Interpret/Display Controls field in Set-Up; it is not available via an escape sequence and cannot be invoked from the host computer.

When the terminal is in a VT200 mode, and you invoke the Set-Up "Display Controls" field, C0, GL, C1, and GR are temporarily loaded as shown in Table 2-6. All characters are displayed in the font shown for C0, GL, C1, and GR.

When the terminal is in a VT52 or VT100 mode, and you invoke the Set-Up "Display Controls" field, C0 and GL are temporarily loaded as shown in Table 2-6. All characters are displayed in the font shown for C0 and GL (C1 and GR are meaningless in VT52 or VT100 modes).

When the terminal is in Display Controls mode, all control functions are displayed and most are prevented from being executed. The only exceptions are that LF, FF, and VT cause a new line (CRLF), and XOFF (DC3) and XON (DC1) maintain flow control if enabled. LF, FF, and VT are displayed before CRLF is executed, and DC1 and DC3 are displayed after execution.

Table 2-6 Display Controls Font (Sheet 1 of 2)

	COLUMN	0		1		2		3		4		5		6		7	
	b8 BITS b7 b6 b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NU	0 0 0	DL	20 16 10		40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SH	1 1 1	D1	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	SX	2 2 2	D2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	EX	3 3 3	D3	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	ET	4 4 4	D4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	EQ	5 5 5	NK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	AK	6 6 6	SY	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BL	7 7 7	EB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	⸮	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	EC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DT	177 127 7F

|←— C0 CODES —→|←— GL CODES (ASCII GRAPHICS) —→|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893D-83

Table 2-6 Display Controls Font (Sheet 2 of 2)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
8 0	200 128 80	9 0	220 144 90	A 0	240 160 A0	°	260 176 B0	À	300 192 C0	D 0	320 208 D0	à	340 224 E0	F 0	360 240 F0	0 0 0 0	0
8 1	201 129 81	9 1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
8 2	202 130 82	9 2	222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
8 3	203 131 83	9 3	223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
8 4	204 132 84	9 4	224 148 94	A 4	244 164 A4	B 4	264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
8 5	205 133 85	9 5	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
8 6	206 134 86	9 6	226 150 96	A 6	246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
8 7	207 135 87	9 7	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
8 8	210 136 88	9 8	230 152 98	¤	250 168 A8	B 8	270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
8 9	211 137 89	9 9	231 153 99	©	251 169 A9	1	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
8 A	212 138 8A	9 A	232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
8 B	213 139 8B	9 B	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
8 C	214 140 8C	9 C	234 156 9C	A C	254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
8 D	215 141 8D	9 D	235 157 9D	A D	255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
8 E	216 142 8E	9 E	236 158 9E	A E	256 174 AE	B E	276 190 BE	Î	316 206 CE	D E	336 222 DE	î	356 238 EE	F E	376 254 FE	1 1 1 0	14
8 F	217 143 8F	9 F	237 159 9F	A F	257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	□	377 255 FF	1 1 1 1	15

|←— C1 CODES —→|←— GR CODES (DEC SUPPLEMENTAL GRAPHICS) —→|

MA 0894A 83

CHAPTER 3
TRANSMITTED CODES

3.1 GENERAL

This chapter describes the codes that the terminal transmits to a program, and presumes that you are familiar with the character encoding concepts described in Chapter 2.

Key codes generated in VT52 mode are listed if they differ from those generated in the ANSI-compatible (VT2ØØ and VT1ØØ) modes.

The terminal can use any of fifteen national keyboards. This chapter describes significant differences.

3.2 KEYBOARD CODES GENERATED

The terminal keyboard (Figure 3-1) consists of: a main keypad, an editing keypad, an auxiliary keypad, and the top-row function keys.

3.2.1 Main Keypad

The main keypad consists of standard keys (used to generate letters, numbers, and symbols) and function keys (used to generate special function codes).

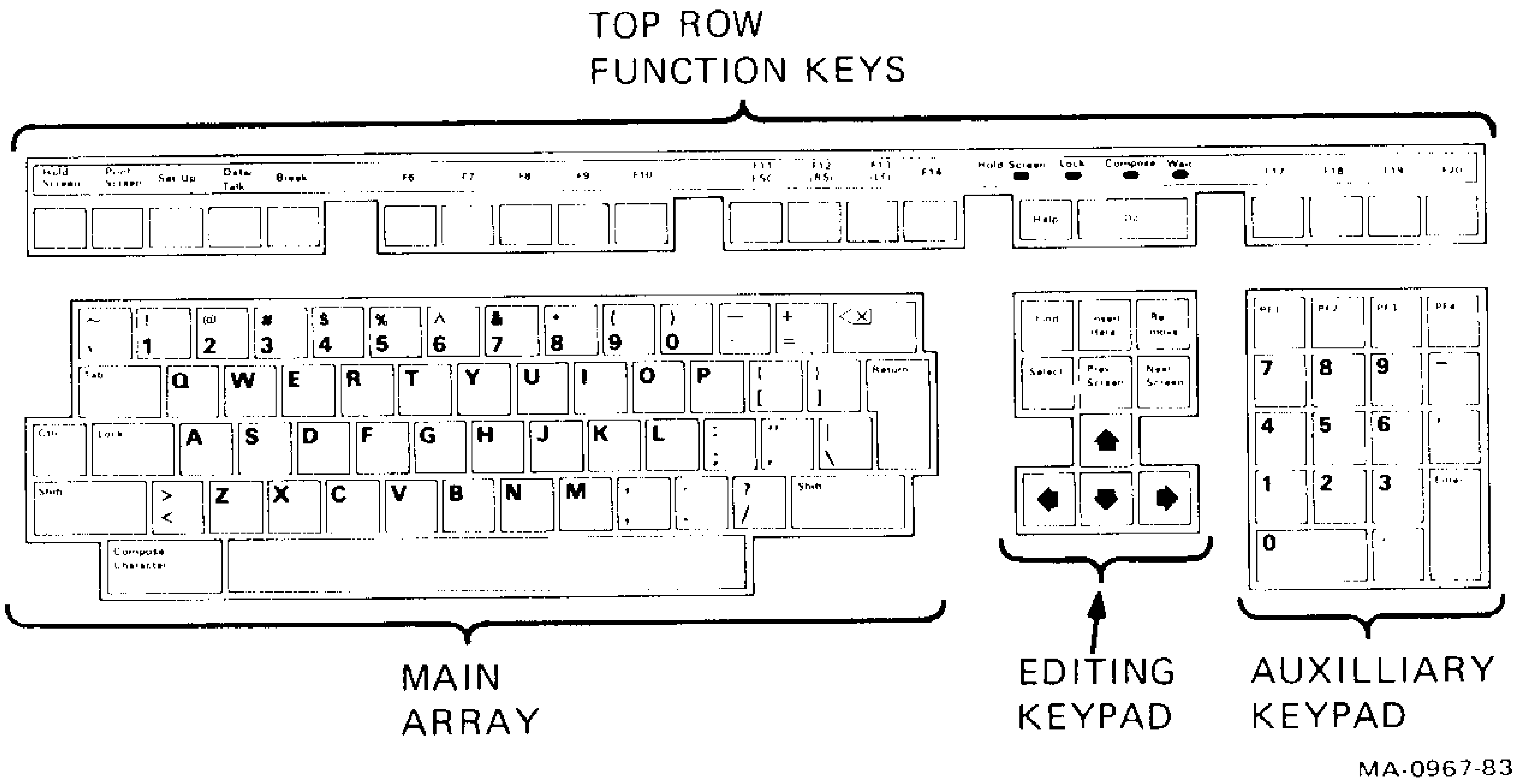


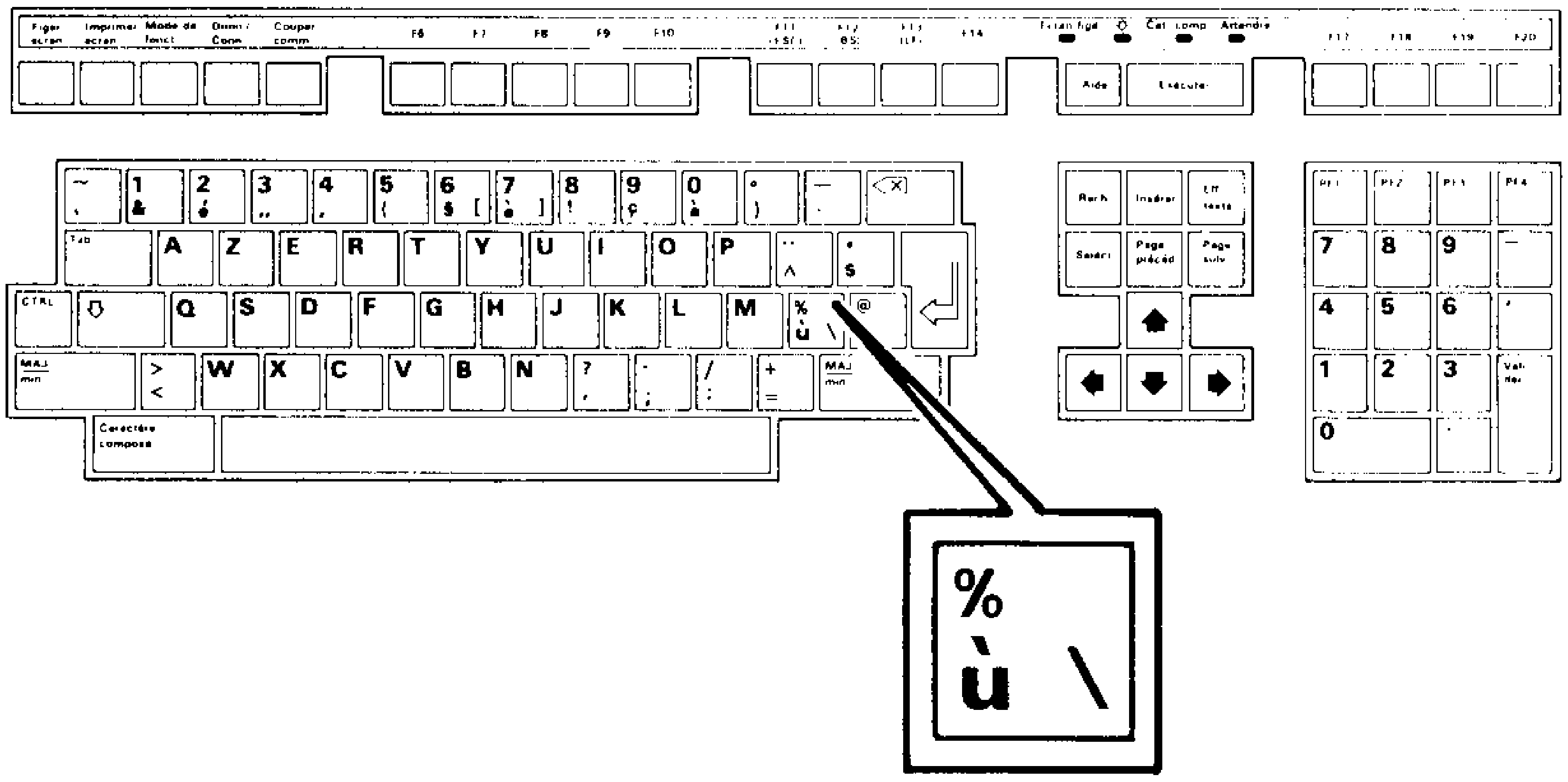
Figure 3-1 Key Groupings (North American Keyboard)

3.2.1.1 **Standard Keys** -- The standard keys generate alphanumeric characters either singly or in combination with other keys.

On the North American keyboard, the standard keys show only ASCII characters. There are no DEC Supplemental characters among the standard keys. Thus, only ASCII codes are generated by these keys. This is a special case, since most keyboards have some standard keys that generate DEC Supplemental as well as ASCII characters. The standard-key patterns vary among keyboards, and some graphic characters (either special symbols or characters with diacritical marks) may or may not be available as standards keys on particular keyboards. Any DEC Multinational graphic character that is not available through a standard key, however, may be created on any keyboard using a compose sequence.

Regardless of which keyboard is used and how a graphic character is created, each character is represented by a unique code according to the character's position in the code table. All GL characters can be used in both 7-bit (VT52 mode, VT1ØØ mode, or 7-bit host line) and 8-bit (VT2ØØ mode, 8-bit host line) environments, while GR characters can only be used in an 8-bit environment.

All keyboards, except the North American Keyboard, have one or more standard keys that transmit different graphic characters (and corresponding codes), depending on whether Typewriter or Data Processing mode has been selected in Set-up (see Figure 3-2 for an example). All keys affected by Typewriter or Data Processing mode have more than two characters shown on their keycaps. The character on the right side of the keycap is generated when the terminal is in Data Processing mode, and the character on the left side of the keycap is generated when the terminal is in Typewriter mode. You can select either shifted (upper) or unshifted (lower) character codes for these keys in the same way as for other nonalphabetic keys.



MA-0968-83

Figure 3-2 French/Belgian Keyboard

3.2.1.2 **Function Keys** -- The main keypad function keys are described as follows. The column/row notation used in the descriptions refer to the Table 2-1 in Chapter 2.

Key — Function

⌫ — The ⌫ (DELETE) key transmits a DEL character (7/15).

TAB — The TAB key transmits an HT character (Ø/9)

RETURN — The RETURN key transmits either a CR character (Ø/13) or a CR character (Ø/13) and an LF character (Ø/1Ø), depending on the set/reset state of Line Feed/New Line mode (LNM) (see Chapter 4).

CTRL — The CTRL key itself does not transmit a code. It is always used in combination with another key to transmit a control code.

LOCK — The LOCK key itself does not transmit a code. It is used in conjunction with the Caps/Shift Lock Mode that is selected in Set-up. It is used to set or clear the CAPS LOCK (or SHIFT LOCK) state.

SHIFT (2 keys) — The SHIFT key itself does not transmit a code. It is used in combination with other standard keys to generate uppercase characters, and is also used in combination with other keyboard keys to generate the "top half" of keys with different top and bottom halves.

SPACE BAR — The space bar transmits a SP character (2/Ø).

COMPOSE CHARACTER — The COMPOSE CHARACTER key does not transmit a code. Pressing the COMPOSE CHARACTER key starts a compose sequence which is used to generate characters that cannot be typed directly from the keyboard.

3.2.2 Editing Keypad

The editing keypad provides editing keys and cursor control keys. Table 3-1 defines the codes generated by the editing keys, and Table 3-2 defines the codes generated by the cursor control keys.

Table 3-1 Codes Generated by Editing Keys

Key	Code Generated VT200 Mode	VT100, VT52 Modes
FIND	9/11 3/1 7/14 CSI 1 ~	--
INSERT HERE	9/11 3/2 7/14 CSI 2 ~	--
REMOVE	9/11 3/3 7/14 CSI 3 ~	--
SELECT	9/11 3/4 7/14 CSI 4 ~	--
PREV SCREEN	9/11 3/5 7/14 CSI 5 ~	--
NEXT SCREEN	9/11 3/6 7/14 CSI 6 ~	--

In VT100 or VT52 modes the editing keys do not generate codes.

Table 3-2 Codes Generated by Cursor Control Keys

	ANSI Mode*		VT52 Mode*	
Key	Cursor Key Mode Reset Normal	Cursor Key Mode Set (Application)	Normal	Application
↑	9/11 4/1 CSI A	8/15 4/1 SS3 A	1/11 4/1 ESC A	1/11 4/1 ESC A
↓	9/11 4/2 CSI B	8/15 4/2 SS3 B	1/11 4/2 ESC B	1/11 4/2 ESC B
→	9/11 4/3 CSI C	8/15 4/3 SS3 C	1/11 4/3 ESC C	1/11 4/3 ESC C
←	9/11 4/4 CSI D	8/15 4/4 SS3 D	1/11 4/4 ESC D	1/11 4/4 ESC D

* ANSI mode applies to VT200 and VT100 modes. VT52 mode is ANSI-incompatible mode.

3.2.3 Auxiliary Keypad

The characters generated by the auxiliary keypad keys depend on the selection of two features: ANSI(VT1ØØ,VT2ØØ)/VT52 and Application keypad features. The Application keypad feature is usually selected only by the computer, but can be selected in Set-up. Refer to Chapter 4 for more information about keypad character selection.

Table 3-3 lists the character codes generated by the auxiliary keypad in ANSI (VT1ØØ, VT2ØØ) mode and in VT52 mode.

Table 3-3 Codes Generated by Auxiliary Keypad Keys

	VT1ØØ/VT2ØØ ANSI Mode*		VT52 Mode*	
Key	Keypad Numeric Mode	Keypad Application Mode	Keypad Numeric Mode	Keypad Application Mode
Ø	3/Ø Ø	8/15 7/Ø SS3 p	3/Ø Ø	1/11 3/15 7/Ø ESC ? p
1	3/1 1	8/15 7/1 SS3 q	3/1 1	1/11 3/15 7/1 ESC ? q
2	3/2 2	8/15 7/2 S3 r	3/2 2	1/11 3/15 7/2 ESC ? r
3	3/3 3	8/15 7/3 SS3 s	3/3 3	1/11 3/15 7/3 ESC ? s
4	3/4 4	8/15 7/4 SS3 t	3/4 4	1/11 3/15 7/4 ESC ? t
5	3/5 5	8/15 7/5 SS3 u	3/5 5	1/11 3/15 7/5 ESC ? u
6	3/6 6	8/15 7/6 SS3 v	3/6 6	1/11 3/15 7/6 ESC ? v
7	3/7 7	8/15 7/7 SS3 w	3/7 7	1/11 3/15 7/7 ESC ? w
8	3/8 8	8/15 7/8 SS3 x	3/8 8	1/11 3/15 7/8 ESC ? x
9	3/9 9	8/15 7/9 SS3 y	3/9 9	1/11 3/15 7/9 ESC ? y
-	2/13 -(minus)	8/15 6/13 SS3 m	2/13 -	1/11 3/15 6/13† ESC ? m
,	2/12 ,(comma)	8/15 6/12 SS3 l	2/12 ,	1/11 3/15 6/12† ESC3 ? l

* ANSI mode applies to VT2ØØ and VT1ØØ modes. VT52 mode is an ANSI-incompatible mode.

† You cannot generate these sequences on a VT52 terminal.

‡ Keypad Numeric Mode. ENTER generates the same codes as RETURN. You can change the code generated by RETURN with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode causes RETURN to generate a single control character (CR). When set, the mode causes RETURN to generate two control characters (CR, LF).

Table 3-3   Codes Generated by Auxiliary Keypad Keys (Cont)

	VT1ØØ/VT2ØØ ANSI Mode*		VT52 Mode*	
Key	Keypad Numeric Mode	Keypad Application Mode	Keypad Application Mode	Keypad Application Mode
.	2/14 .(period)	8/15 6/14 SS3 n	2/14 .	1/11 3/15 6/14 ESC ? n
Enter‡	Ø/13 CR or Ø/13 Ø/1Ø CR LF	8/15 4/13 SS3 M	Ø/13 CR or Ø/13 Ø/1Ø CR LF	1/11 3/15 4/13 ESC ? M
PF1	8/15 5/1Ø SS3 P	8/15 5/Ø SS3 P	1/11 5/Ø ESC P	1/11 5/Ø ESC P
PF2	8/15 5/1 SS3 Q	8/15 5/1 SS3 Q	1/11 5/1 ESC Q	1/11 5/1 ESC Q
PF3	8/15 5/2 SS3 R	8/15 5/2 SS3 R	1/11 5/2 ESC R	1/11 5/2 ESC R
PF4	8/15 5/3 SS3 S	8/15 5/3 SS3 S	1/11 5/3 ESC S	1/11 5/3† ESC S

* ANSI mode applies to VT2ØØ and VT1ØØ modes. VT52 mode is an ANSI-incompatible mode.

† You cannot generate these sequences on a VT52 terminal.

‡ In Keypad Numeric Mode, ENTER generates the same codes as RETURN. You can change the code generated by RETURN with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode causes RETURN to generate a single control character (CR). When set, the mode causes RETURN to generate two control characters (CR, LF).

3.2.4 Top-Row Function Keys

There are twenty top-row function keys generically named F1 through F2Ø. The first five keys (F1 through F5) labeled HOLD SCREEN, PRINT SCREEN, SET-UP, DATA/TALK, and BREAK do not transmit codes; they are local function keys. Keys F6 through F2Ø, however, transmit the codes defined in Table 3-4.

Table 3-4   Codes Generated by Top Row Function Keys

		Code Generated				
Name on Legend Strip	Generic Name	VT2ØØ Mode				VT1ØØ, VT52 Modes
HOLD SCREEN	(F1)*	--				--
PRINT SCREEN	(F2)*	--				--
SET-UP	(F3)*	--				--
DATA/TALK	(F4)*	--				--
BREAK	(F5)*	--				--
F6	F6	9/11 CSI	3/1 1	3/7 7	7/14 ~	--
F7	F7	9/11 CSI	3/1 1	3/8 8	7/14 ~	--
F8	F8	9/11 CSI	3/1 1	3/9 9	7/14 ~	--
F9	F9	9/11 CSI	3/2 2	3/Ø Ø	7/14 ~	--
F1Ø	F1Ø	9/11 CSI	3/2 2	3/1 1	7/14 ~	--
F11 (ESC)	F11	9/11 CSI	3/2 2	3/3 3	7/14 ~	1/11 ESC
F12(BS)	F12	9/11 CSI	3/2 2	3/4 4	7/14 ~	Ø/8 BS
F13(LF)	F13	9/11 CSI	3/2 2	3/5 5	7/14 ~	Ø/1Ø LF
F14	F14	9/11 CSI	3/2 2	3/6 6	7/14 ~	--
HELP	(F15)	9/11 CSI	3/2 2	3/8 8	7/14 ~	--
DO	(F16)	9/11 CSI	3/2 2	3/9 9	7/14 ~	--
F17	F17	9/11 CSI	3/3 3	3/1 1	7/14 ~	--

* F1 through F5 are local function keys and do not generate codes.

Table 3-4   Codes Generated by Top Row Function Keys (Cont)

		Code Generated	
Name on Legend Strip	Generic Name	VT2ØØ Mode	VT1ØØ, VT52 Modes
F18	F18	9/11 3/3 3/2 7/14 CSI 3 2 ~	--
F19	F19	9/11 3/3 3/3 7/14 CSI 3 3 ~	--
F2Ø	F2Ø	9/11 3/3 3/4 7/14 CSI 3 4 ~	--

3.2.5 Control Codes Generated

Table 3-5 defines the keys and key combinations used to generate control codes. These keys and combinations are valid on all keyboards. These control codes are CØ 7-bit control characters; there is no similar mechanism for generating Cl 8-bit control characters.

3.3 ENABLING AND DISABLING AUTOREPEAT

Autorepeat can be enabled or disabled through the keyboard using Set-up or using the DECARM sequence (Chapter 4). If the escape sequence to turn auto repeat off (DECARM) is received while an auto repeat is in progress, the key stops auto repeating. If the escape sequence to turn auto repeat on is received when a key which has been auto repeating is still held down, the key will immediately auto repeat without delay. Keys which can autorepeat will normally start autorepeating after a delay of 1/2 second.

To give a constant repeat rate at all transmission speeds, the speed of auto repeat is a function of the host transmission speed. At speeds of 24ØØ baud or above, all keys auto repeat at 3Ø keystrokes per second. For the purpose of auto repeat, the keyboard is separated into the following three groups:

Group A -- Main typing array
Group B -- Cursor keys and keypad keys
Group C -- Top row function keys and editing keys

Every key in each group auto-repeats at the fixed rate set by the transmit speed regardless of how many codes the key actually transmits.

Table 3-5 Keys Used to Generate 7-Bit Control Characters

Control Character Mnemonic	Code	Key Pressed with CTRL (All Modes)	Dedicated Function Key
NUL	Ø/ØØ	2, space	
SOH	Ø/Ø1	A	
STX	Ø/Ø2	B	
ETX	Ø/Ø3	C	
EOT	Ø/Ø4	D	
ENQ	Ø/Ø5	E	
ACK	Ø/Ø6	F	
BEL	Ø/Ø7	G	
BS	Ø/Ø8	H	F12 (BS)*
HT	Ø/Ø9	I	TAB
LF	Ø/1Ø	J	F13 (LF)*
VT	Ø/11	K	
FF	Ø/12	L	
CR	Ø/13	M	RETURN
SO	Ø/14	N	
SI	Ø/15	O	
DLE	1/ØØ	P	
DC1	1/Ø1	Q**	
DC2	1/Ø2	R	
DC3	1/Ø3	S**	
DC4	1/Ø4	T	
NAK	1/Ø5	U	
SYN	1/Ø6	V	
ETB	1/Ø7	W	
CAN	1/Ø8	X	
EM	1/Ø9	Y	
SUB	1/1Ø	Z	
ESC	1/11	3,[	F11 (ESC)*
FS	1/12	4,/	
GS	1/13	5,]	
RS	1/14	6,~	
US	1/15	7,?	
DEL	7/15	8	DELETE

* Keys F11, F12, and F13 generate these 7-bit control characters only when the terminal is operated in VT1ØØ mode or VT52 mode.

** These keystrokes are enabled only if XOFF support is disabled. If XOFF support is enabled, then CTRL-S is a "hold screen" local function and CTRL-Q is an "unhold screen" local function.

Auto Repeat Rate

Host Port Transmit Speed	Group A	Group B	Group C
>=2400 Baud	30 keys/sec	30 keys/sec	30 keys/sec
1200	30	30	24
600	30	20	12
300	30	12	12
150	6	6	6
110	6	6	6
75	6	6	6

The "Transmit Rate Limit" feature (selectable in Set-up) does not, in general, affect auto repeat rates since all five codes can be transmitted at the limited speed of 150 characters per second at most baud rates. In local mode, keys will auto repeat at 30 keystrokes per second.

The following keys do not auto repeat: HOLD SCREEN, PRINT SCREEN, SET-UP, DATA/TALK, BREAK, COMPOSE CHARACTER, SHIFT, RETURN, LOCK, and CTRL. Shifted or controlled keys will auto repeat.

3.4 KEYBOARD LOCK AND UNLOCK

The keyboard can lock as result of the following conditions:

- If the program sends a command to set the keyboard Action Mode (KAM) (see Chapter 4).

- If the keyboard output buffer is full.

When the keyboard is locked, all keyboard keys except HOLD SCREEN, PRINT SCREEN, SET-UP, DATA/TALK and BREAK are disabled, and the keyboard WAIT indicator lights up.

If the keyboard is locked, it can be unlocked as a result of any of the following:

- Reduction of output buffer contents to less than full (assuming KAM is not set).

- Receipt of KAM reset sequence if buffer is not full (see Chapter 4). Also both DECSTR and Reset Terminal reset KAM.

- Invoking Clear Comm, Reset Terminal, Recall, or Default functions using Set-Up. (Entering Set-Up unlocks the keyboard for the time the terminal is in Set-up. If these functions are not invoked while in Set-up, the keyboard locks again when Set-up is exited).

- Performing the Power-Up Self Test (DECTST) or RIS.

CHAPTER 4
RECEIVED CODES

4.1 GENERAL

This chapter describes the terminal's response to codes it may receive from an application or host system. This Chapter assumes you are familiar with the character encoding conventions and terminology covered in Chapter 2.

All data received by the VT22Ø consists of single and multiple-character codes: graphic (printing or display) characters, control characters, escape sequences, control sequences and device control strings. Much of that data consists of graphic characters that simply appear on the screen with no other effect. Control characters, escape sequences, control sequences and device control strings are all "control functions" that you use in your program to specify how the terminal should process, transmit, and display characters. Each control function has a unique name and each name has a unique abbreviation (mnemonic). Both the name and the abbreviation are standardized.

By default, the terminal interprets individual control and graphic characters according to the DEC Multinational Character Set code mapping (Chapter 2).

NOTE: The terminal usually ignores control codes it does not support. However, codes sent to the terminal other than those specified in this manual can cause unexpected results.

The codes described in this chapter are described as used in VT2ØØ mode unless otherwise specified.

4.2 CONTROL CHARACTERS

Tables 4-1 and 4-2 define the action taken by the terminal when receiving CØ and C1 control characters. The VT22Ø does not recognize all CØ or C1 characters. Those not shown in either table are ignored (no action taken).

Table 4-1 CØ (ASCII) Control Characters Recognized

Mnemonic	Code	Name	Action Taken
NUL	Ø/Ø	Null	Ignored when received.
ENQ	Ø/5	Enquiry	Answerback message is generated.
BEL	Ø/7	Bell	Generates bell tone if bell is enabled.
BS	Ø/8	Backspace	Moves cursor to the left one character position: if cursor is at left margin, no action occurs.
HT	Ø/9	Horizontal tabulation	Moves cursor to next tab stop, or to right margin if there are no more tab stops. Does not cause autowrap.
LF	Ø/1Ø	Linefeed	Causes a linefeed or a new line operation, depending on the setting of new line mode.
VT	Ø/11	Vertical tabulation	Processed as LF.
FF	Ø/12	Form feed	Processed as LF.
CR	Ø/13	Carriage return	Moves cursor to left margin on current line.
SO (LS1)	Ø/14	Shift out (Lock shift G1)	Invokes G1 character set into GL. G1 is designated by a select-character-set (SCS) sequence.
SI (LSØ)	Ø/15	Shift in (Lock shift GØ)	Invoke GØ character set into GL. GØ is designated by a select-character-set sequence (SCS).
DC1	1/1	Device Control 1	Also referred to as XON. If XOFF support is enabled, DC1 clears DC3 (XOFF), causing the terminal to continue transmitting characters (keyboard unlocks) unless KAM mode is currently set.

Table 4-1 CØ (ASCII) Control Characters Recognized (Cont)

Mnemonic	Code	Name	Action Taken
DC3	1/3	Device Control 3	Also referred to as XOFF. If XOFF support is enabled, DC3 causes the terminal to stop transmitting characters until a DC1 control character is received.
CAN	1/8	Cancel	If received during an escape or control sequence, terminates and cancels the sequence. No error character is displayed. If received during a device control string, the DCS is terminated and no error character is displayed.
SUB	1/1Ø	Substitute	If received during escape or control sequence, terminates and cancels the sequence. Causes a reverse question mark to be displayed. If received during a device control sequence, the DCS is terminated and reverse question mark is displayed.
ESC	1/11	Escape	Processed as escape sequence introducer. Terminates any escape, control or device control sequence which is in progress.
DEL	7/15	Delete	Ignored when received. Note: May not be used as a time fill character.

Table 4-2 C1 Control Characters Recognized

Mnemonic	8-Bit Code	Equivalent 7-Bit Code Extension		Name	Action Taken
IND	8/4	1/11 ESC	4/4 D	Index	Moves cursor down one line in same column. If cursor is at bottom margin, screen performs a scroll up.
NEL	8/5	1/11 ESC	4/5 E	Next line	Moves cursor to first position on next line. If cursor is at bottom margin, screen performs a scroll up.
HTS	8/8	1/11 ESC	4/8 H	Horizontal tab set	Sets one horizontal tab stop at the column where the cursor is.
RI	8/13	1/11 ESC	4/13 M	Reverse index	Moves cursor up one line in same column. If cursor is at top margin, screen performs a scroll down.
SS2	8/14	1/11 ESC	4/14 N	Single shift G2	Temporarily invokes G2 character set into GL for the next graphic character. G2 is designated by a select-character-set (SCS) sequence.

Table 4-2 C1 Control Characters Recognized (Cont)

Mnemonic	8-Bit Code	Equivalent 7-Bit Code Extension		Name	Action Taken
SS3	8/15	1/11 ESC	4/15 O	Single shift G3	Temporarily invokes G3 character set into GL for the next graphic character. G3 is designated by a select-character-set (SCS) sequence.
DCS	9/Ø	1/11 ESC	5/Ø P	Device control string	Processed as opening delimiter of a device control string for device control use.
CSI	9/11	1/11 ESC	5/11 [	Control sequence introducer	Processed as control sequence introducer.
ST	9/12	1/11 ESC	5/12 \	String terminator	Processed as closing delimiter of a string opened by DCS.

Table 4-1 shows that SO (Ø/14) and SI (Ø/15) are also called LS1 and LSØ, respectively. SO and SI (shift out and shift in) are the traditional ASCII names or mnemonics. LS1 and LSØ (lock shift G1 and lock shift GØ) are alternate names that are useful when dealing with the variety of character set mappings possible. LS1 and LSØ are the names used in this chapter.

Table 4-2 shows the equivalent 7-bit code extension for each 8-bit C1 code. The code extensions require one more byte than the C1 codes. Chapter 2 describes when to use C1 codes and when to use 7-bit code extensions. The "code extension announcers" described in Chapter 2 are also listed in this chapter.

4.3 COMPATIBILITY LEVEL (DECSCL)

The terminal can be set for a particular level of operation for interface compatibility. There are two possible levels: Level 1 (VT1ØØ operation) and Level 2 (VT2ØØ operation). The VT22Ø (a Level 2 terminal) can be set for either Level 1 or Level 2. These functional levels are defined in Table 4-3.

Table 4-3 Level 1-Level 2 Compatibility Comparison

Difference Area	Level 1 (VT1ØØ Mode)	Level 2 (VT2ØØ Mode)
Keyboard	Sends ASCII only Keystrokes that normally send DEC Supplemental Characters transfer nothing (interpreted as as error) User-defined keys are inoperative Special function keys and six editing keys are inoperative (except that F11, F12, and F13 transmit ESC, BS, and LF, respectively)	Full use of VT22Ø keyboard
7 or 8 bits	The 8th bit of all received characters is set to zero (Ø)	The 8th bit of all received characters is significant
Character Sets	ASCII, UK, and Special Graphics only	All VT22Ø Character Sets (except U.K.) are available
C1 Control Characters	All transmitted C1 controls are forced to S7C1 state and transmitted as 7-bit escape sequences	

You can set the compatibility level of the terminal using the following sequences.

NOTE: Restrictions for a lower compatibility level need only be applied when there is a non-compatible interaction with software written for the lower level.

Sequence	Action
9/11 3/6 3/1 2/2 7/0 CSI 6 1 " p	Set terminal for level 1 compatibility (VT100 mode).
9/11 3/6 3/2 2/2 7/0 CSI 6 2 " p	Set terminal for level 2 compatibility (VT200 mode, 8-bit controls).
9/11 3/6 3/2 3/11 3/0 2/2 7/0 CSI 6 2 ; 0 " p	Set terminal for level 2 compatibility (VT200 mode, 8-bit controls).
9/11 3/6 3/2 3/11 3/1 2/2 7/0 CSI 6 2 ; 1 " p	Set terminal for level 2 compatibility (VT200 mode, 7-bit controls).
9/11 3/6 3/2 3/11 3/2 2/2 7/0 CSI 6 2 ; 2 " p	Set terminal for level 2 compatibility (VT200 mode, 8-bit controls).

4.4 CHARACTER SET SELECTION (SCS)

Character encoding in the VT220 was introduced in Chapter 2. The control functions you need to select different graphic character sets are described below. Differences with a VT100 terminal are pointed out where they may affect software compatibility between the VT220 and VT100-type terminals.

The VT220's graphic character repertoire consists of the following graphic sets (see Chapter 2 for a description of these character sets):

- ASCII Graphics
- DEC Supplemental Graphics
- DEC Special Graphics
- UK National
- Down-line Loadable

Generally, you select character sets as follows (Figure 4-1).

First, using SCS sequences, you "designate" graphic sets as G0, G1, G2, G3. This makes the graphics sets available "on call" for your program. To actually map any one of these sets into GL or GR you must then "invoke" any of G0 through G3 into GL or GR by using locking shifts (LS0, LS1, LS2, LS1R, LS2R, LS3R) or single shifts (SS2, SS3).

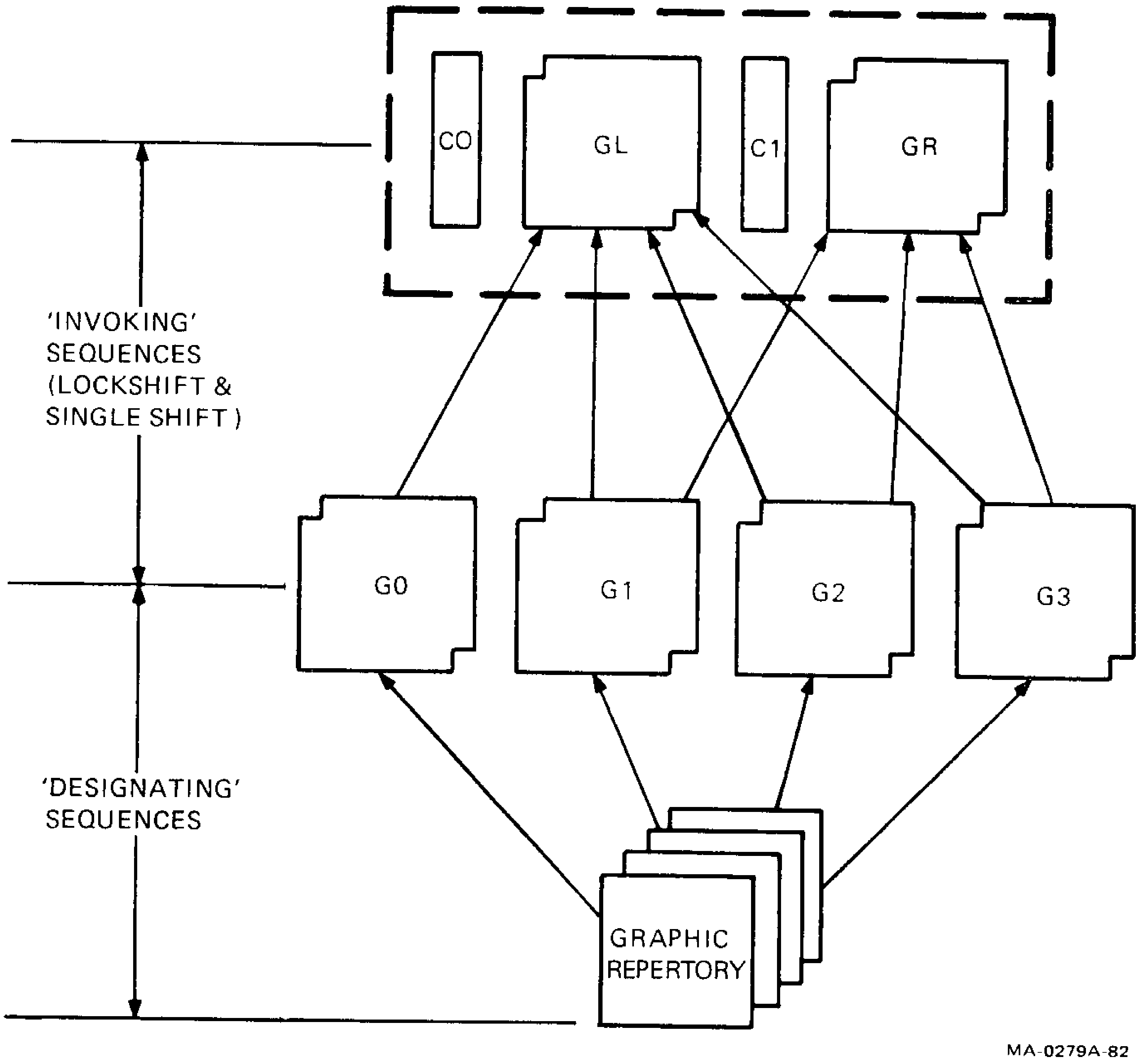


Figure 4-1 Character Set Selection

Character sets remain designated until the terminal receives another SCS sequence. All locking shifts remain active until the terminal receives another locking shift. Single shifts SS2 and SS3 remain active for only the next single graphic character.

You do not need to select character sets in this manner every time you use the terminal because there is a default mapping. The default mapping in VT200 mode is: ASCII Graphics in GL and DEC Supplemental Graphics in GR (DEC Multinational). The default graphic character set mapping is reset whenever you power up the terminal. Your application can also select the default mapping by means of the soft terminal reset sequence (DECSTR).

All control functions for designating and invoking graphic character sets are described below.

4.4.1 Designating "Hard" Character Sets

You designate "hard" character sets (ASCII, U.K. National, DEC Supplemental, and DEC Special Graphics) using the escape sequences in Table 4-4.

Table 4-4 Designating "Hard" Character Sets

Character Set	Escape Sequence	Designate as:
ASCII	1/11 2/8 4/2 ESC (B	GØ (default)
	1/11 2/9 4/2 ESC) B	G1
	1/11 2/1Ø 4/2 ESC * B	G2 (VT2ØØ mode only)
	1/11 2/11 4/2 ESC + B	G3 (VT2ØØ mode only)
DEC Supplemental (VT2ØØ mode only)	1/11 2/8 3/12 ESC (<	GØ
	1/11 2/9 3/12 ESC) <	G1
	1/11 2/1Ø 3/12 ESC * <	G2
	1/11 2/11 3/12 ESC + <	G3
U.K. National	1/11 2/8 4/1 ESC (A	GØ
(VT1ØØ mode only)	1/11 2/9 4/1 ESC) A	G1
DEC Special Graphics	1/11 2/8 3/Ø ESC (Ø	GØ
	1/11 2/9 3/Ø ESC) Ø	G1
	1/11 2/1Ø 3/Ø ESC * Ø	G2 (VT2ØØ mode only)
	1/11 2/11 3/Ø ESC + Ø	G3 (VT2ØØ mode only)

4.4.2 Designating "Soft" (Down-Line Loadable) Character Sets

It is possible for you to define a soft character set that may or may not replace one of the existing "hard" sets (ROM based fonts). If you do replace a "hard" set, the replacement occurs for both the 80 and 132 column versions. A soft font which replaces a hard font remains in effect until the soft font is cleared or redefined. The soft font is cleared by RECALL, DEFAULT, self test, power up, and is redefined by DECDLD. If the soft character set you define does not replace an existing hard set, then it is used in addition to the hard sets.

NOTE: Designation of a soft character set is supported only in the VT200 mode.

You designate a soft (down-Line loadable) character set using the following escape sequences:

Escape Sequence	Designate As:
1/11 2/8 .... ESC (Dscs	G0
1/11 2/9 .... ESC) Dscs	G1
1/11 2/10 ...... ESC * Dscs	G2
1/11 2/11 ...... ESC + Dscs	G3

In these sequences, Dscs is a variable that defines the character set for the soft font.

Dscs	Function
I I F	Generic Dscs A Dscs can consist of between 0 and 2 intermediates (I) and a final (F). Intermediates are in the range 2/0-2/15, Finals are in the range 3/0-7/14.

Three examples of Dscs are:

Dscs	Function
2/0 4/0 sp @	Defines character set as unregistered soft set. This value is the recommended default value for user defined sets.
4/2 B	Defines the soft character set to be ASCII.
2/6 2/5 4/3 & % c	Defines character set as "&% c", which is currently an unregistered set.

4.4.3 Invoking Character Sets Using Locking Shifts

Once you have designated your character sets, you can invoke GØ, G1, G2 or G3 into GL or GR by using the Locking Shift control functions as shown in Table 4-5, Figure 4-2, and Figure 4-3.

Table 4-5 Invoking Character Sets Using Lock Shifts

Control Name	Coding		Function
LSØ -- Lock Shift GØ	Ø/15 SI		Invoke GØ into GL. (default)
LS1 -- Lock Shift G1	Ø/14 SO		Invoke G1 into GL.
LS1R -- Lock Shift G1, Right	1/11 ESC	7/14 ~	Invoke G1 into GR. VT2ØØ mode only. Use of this sequence can cause software incompatibility problems.
LS2 -- Lock Shift G2	1/11 ESC	6/14 n	Invoke G2 into GL. VT2ØØ mode only. Use of this sequence can cause software incompatibility problems.
LS2R -- Lock Shift G2, Right	1/11 ESC	7/13 }	Invoke G2 into GR. (default) VT2ØØ mode only.
LS3 -- Lock Shift G3	1/11 ESC	6/15 o	Invoke G3 into GL. VT2ØØ mode only. Use of this sequence can cause software incompatibility problems.
LS3R -- Lock Shift G3, Right	1/11 ESC	7/12 |	Invoke G3 into GR. VT2ØØ mode only.

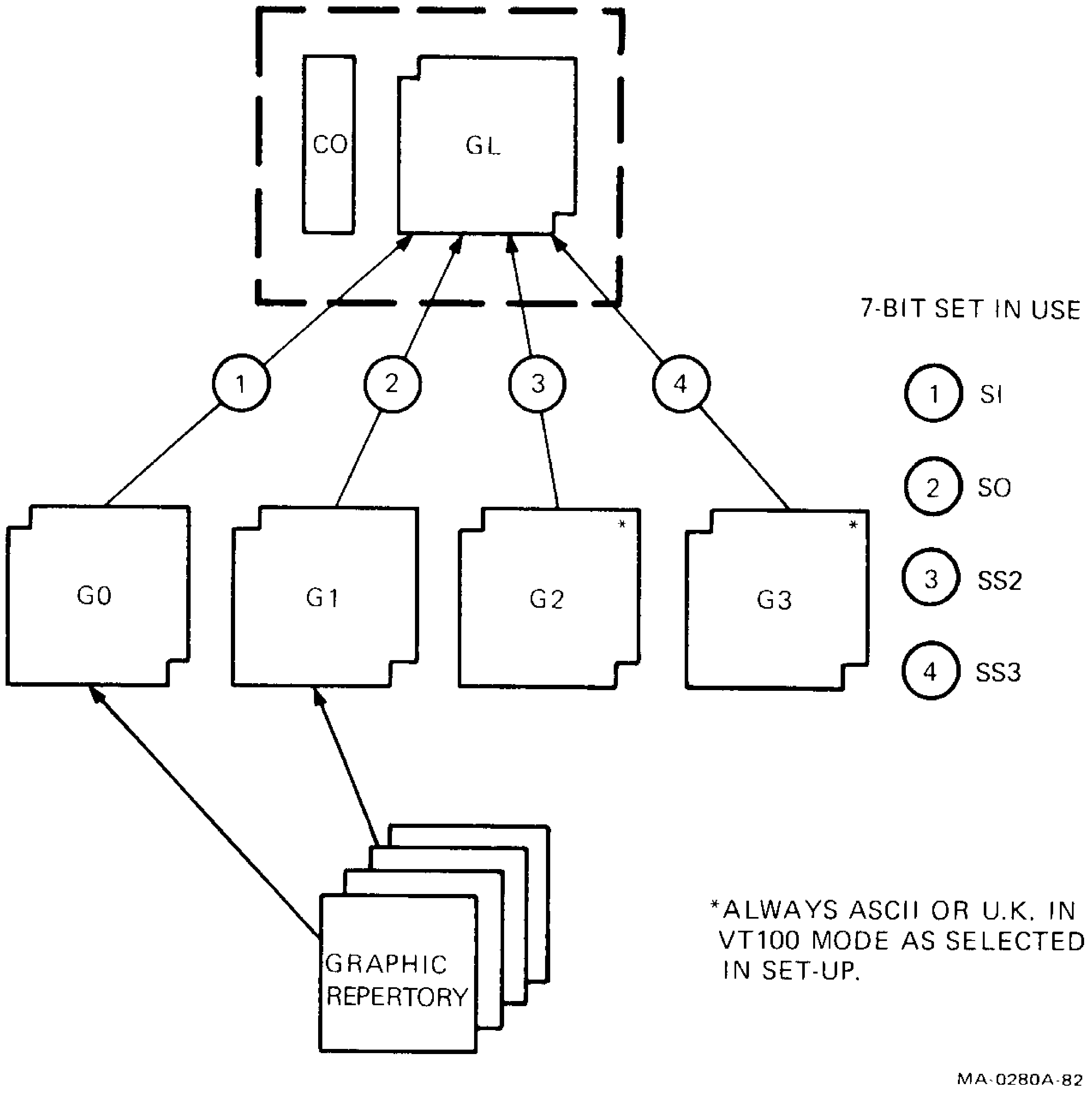


Figure 4-2 Locking and Single-Shift Commands (VT1ØØ Mode)

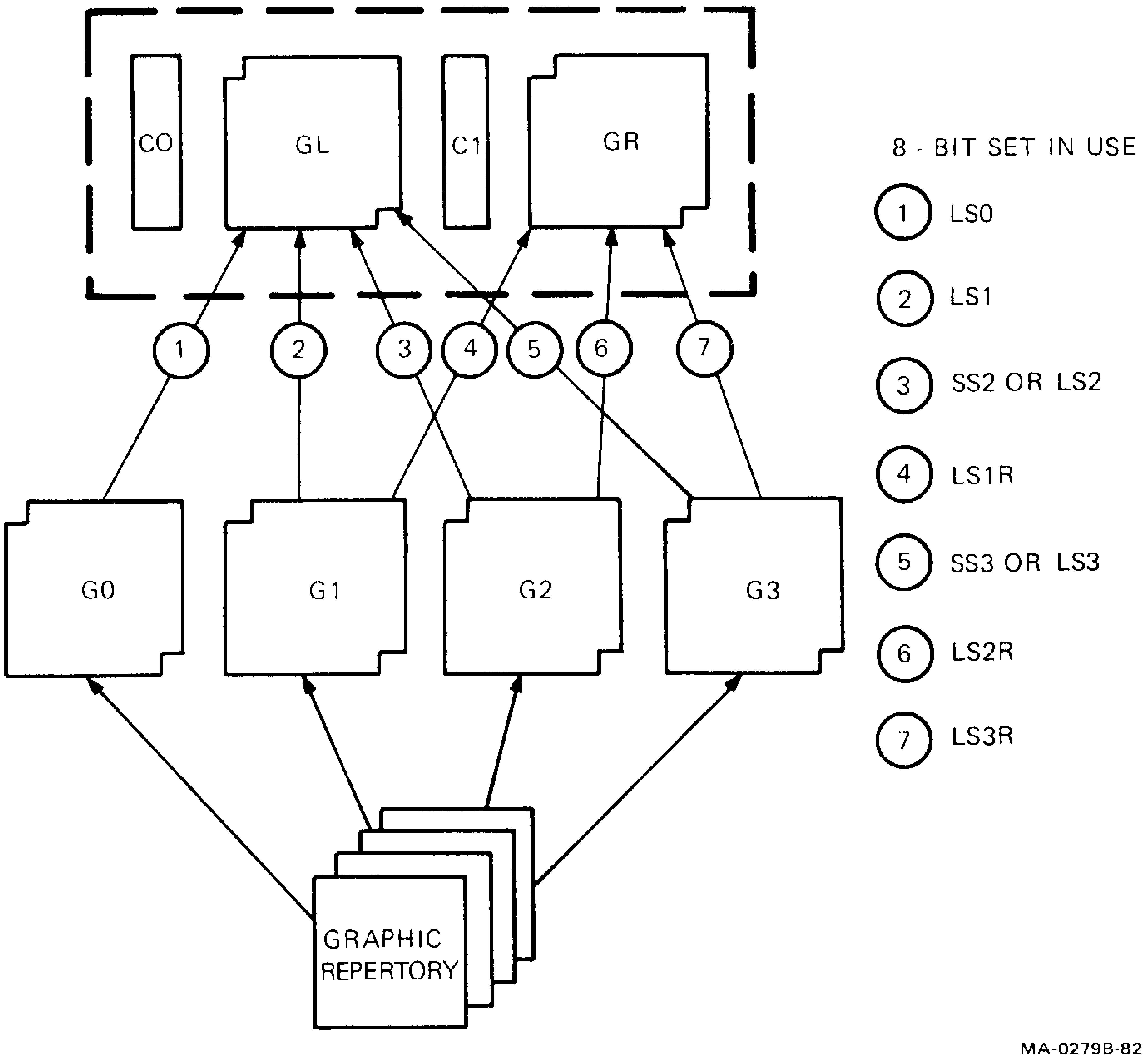


Figure 4-3   Locking and Single-Shift Commands (VT2ØØ Mode)

4.4.4 Invoking Character Sets Using Single Shifts

Once you have designated your character sets, you can invoke G2 or G3 into GL for a single graphic character at a time using the single shift control function described below and in Figures 4-2 and 4-3.

All single shifts remain active for only the next single graphic character. The terminal returns the previous character set after displaying a single graphic character.

4.4.4.1 SS2 -- Single Shift G2 -- SS2 is an 8-bit control character (8/14) that invokes G2 into GL for the next graphic character. You can also express SS2 as an escape sequence when coding for a 7-bit environment:

```
1/11  4/14
ESC    N
```

4.4.4.2 SS3 -- Single Shift G3 -- SS3 is an 8-bit control character (8/15) that invokes G3 into GL for the next graphic character. You can also express SS3 as an escape sequence when coding for a 7-bit environment:

```
1/11  4/15
ESC    O
```

4.5 SELECT C1 CONTROLS

You can use Select C1 Controls (codes extension announcers) in your program to control the representation of C1 control codes returned to the application by the terminal (see Chapter 2 for information on working with 7- and 8-bit environments). The terminal always accepts 7 or 8 bit forms for C1 controls in either of the VT2ØØ modes.

Digital recommends that you use DECSCL sequences instead of Select C1 Controls because DECSCL performs a soft reset putting the terminal in a "known" state in addition to setting the terminal mode and C1 control state.

NOTE: These sequences are supported only in VT2ØØ mode.

4.5.1 Select 7-Bit C1 Control Transmission (S7C1T)

```
1/11  2/Ø 4/7
ESC   sp   F
```

Causes all C1 codes returned to the application to be converted to their equivalent 7-bit code extensions.

NOTE: The S7C1T sequence is ignored when the terminal is in VT1ØØ or VT52 mode.

4.5.2 Select 8-Bit C1 Control Transmission (S8C1T)

```
1/11 2/0  4/6
ESC  sp    G
```

Causes the terminal to return C1 codes to the application without converting them to their equivalent 7-bit code extensions.

4.6 TERMINAL MODES

A mode is a state of the terminal that affects the way the terminal operates. Modes that are selectable, and their set/reset control sequences, are listed in Table 4-6 and described in this section. Each mode has an identifying name (mnemonic), and each can be set or reset individually or in strings using Set Mode (SM) or Reset Mode (RM) control sequences. Certain features, called User Preference features, can be "locked" by the operator using Set-Up; thus preventing the host from changing the feature.

DIGITAL private control sequences (permitted within the extensions of ANSI standards) are identified by DEC in the control sequence mnemonic, which includes a question mark character (?) after the Control Sequence Introducer.

These modes are selectable by the terminal operator using Set-up screens.

4.6.1. Set Mode (SM)

The Set Mode command for ANSI modes is:

```
9/11  3/11  3/11   6/8
CSI   Ps     ; .... ;   Ps h
```

The Set Mode command for DIGITAL private modes is:

```
9/11  3/15  3/11   3/11 6/8
CSI    ?     ; .... ;   Ps h
```

You use these commands to set the ANSI and DIGITAL private modes, individually or in strings, using the parameters (Ps) listed in Tables 4-7 and 4-8. ANSI modes and DIGITAL private modes cannot be used in the same SM string.

Table 4-6 Selectable Modes Summary

Name	Mnemonic	Set Mode	Reset Mode*
Keyboard** Action	KAM	Locked CSI 2 h	Unlocked CSI 2 l
Insertion-Replacement	IRM	Insert CSI 4 h	Replace CSI 4 l
Send-Receive	SRM	Off CSI 12 h	On CSI 12 l
Line Feed-New Line	LNM	New Line CSI 2Ø h	Line Feed CSI 2Ø l
Cursor Key	DECCKM	Application CSI ? 1 h	Cursor CSI ? 1 l
ANSI/VT52	DECANM	N/A CSI ? 2 l	VT52
Column	DECCOLM	132 Column CSI ? 3 h	8Ø Column CSI ? 3 l
Scrolling**	DECSCLM	Smooth CSI ? 4 h	Jump CSI ? 4 l
Screen**	DECSCNM	Reverse CSI ? 5 h	Normal CSI ? 5 l
Origin mode	DECOM	Origin CSI ? 6 h	Absolute CSI ? 6 l
Auto Wrap	DECAWN	On CSI ? 7 h	Off CSI ? 7 l
Auto Repeat**	DECARM	On CSI ? 8 h	Off CSI ? 8 l
Print Form Feed	DECPFF	On CSI ? 18 h	Off CSI ? 18 l
Print Extent	DECPEX	Full Screen CSI ? 19 h	Scrolling Region CSI ? 19 l
Test Cursor Enable	DECTCEM	On CSI ? 25 h	Off CSI ? 25 l
Keypad	DECKPAM DECKPNM	Application ESC =	Numeric ESC >

* The last character of each sequence is lowercase L (6/12)
**User Preference feature

Table 4-7 ANSI-Standardized Modes

Name	Mnemonic	Parameter (Ps)	
Error (ignored)		Ø	(3/Ø)
Keyboard Action	KAM	2	(3/2)
Insertion-replacement	IRM	4	(3/4)
Send-receive	SRM	12	(3/1 3/2)
Linefeed/New Line	LNM	2Ø	(3/2 3/Ø)

Table 4-8 ANSI-Compatible DEC Private Modes

Name	Mnemonic	Parameter (Ps)	
Error (ignored)		Ø	(3/Ø)
Cursor Key	DECCKM	1	(3/1)
ANSI/VT52	DECANM	2	(3/2)
Column	DECCOLM	3	(3/3)
Scroll	DECSCLM	4	(3/4)
Screen	DECSCNM	5	(3/5)
Origin	DECOM	6	(3/6)
Auto Wrap	DECAWM	7	(3/7)
Auto Repeat	DECARM	8	(3/8)
Printer Form Feed	DECPFF	18	(3/1 3/8)
Printer Extent	DECPEX	19	(3/1 3/9)
Text Cursor Enable	DECTCEM	25	(3/2 3/5)

4.6.2 Reset Mode (RM)

The Reset Mode command for ANSI modes is:

```
    9/11          3/11  3/11      6/12
    CSI    Ps      ; ..... ;    Ps 1
```

The Reset Mode command for DIGITAL private modes is:

```
    9/11  3/15  3/11  3/11      6/12
    CSI    ?     ; .... ;     Ps 1
```

You use these commands to reset the ANSI and DIGITAL private modes, individually or in strings, using the parameters (Ps) listed in Tables 4-7 and 4-8. ANSI modes and DIGITAL modes cannot be used in the same RM string.

4.6.3 Keyboard Action Mode (KAM)

Keyboard Action mode lets your program lock and unlock the keyboard. When the keyboard is locked, no codes can be transmitted from the keyboard to the program. To alert the operator, whenever the keyboard is locked, the WAIT indicator lights and the keyclick feature is disabled. You select keyboard action mode by using the following sequences.

NOTE: This is a User Preference feature and can be locked using Set-up.

Mode	**Sequence**	**Action**
Set	9/11 3/2 6/8 CSI 2 h	Locks the keyboard for the next and subsequent keystrokes.
Reset	9/11 3/2 6/12 CSI 2 1	Unlocks the keyboard, unless it is locked by DC3.

4.6.4 Insert/Replacement Mode (IRM)

The terminal displays received characters at the cursor position. Insert/Replacement mode determines how the terminal adds characters to the screen. Insert mode displays the character and moves previously displayed characters to the right. Replace mode adds characters by replacing the character at the cursor position. You select Insert/Replacement Mode by using the following sequences.

Mode	Sequence	Action
Set	9/11 3/4 6/8 CSI 4 h	Selects insert mode. New display characters move old display characters to the right. Characters moved past the right margin are lost.
Reset	9/11 3/4 6/12 CSI 4 l	Selects replace mode. New display characters replace old display characters at the cursor position. The old character is erased.

4.6.5 Send-Receive Mode (SRM)

Send-Receive Mode turns local echo on or off. When send-receive mode is reset (local echo on), every character transmitted from the keyboard automatically appears on the screen. Therefore, the host does not have to transmit (echo) the character back to the terminal display. When send-receive mode is set (local echo off), the terminal transmits characters only to the application. The host must echo the characters back to the terminal display.

Mode	Sequence	Action
Set	9/11 3/1 3/2 6/8 CSI 1 2 h	Turns off (disables) local echo. When the terminal transmits characters to the host, the host must echo characters back to the terminal display.
Reset	9/11 3/1 3/2 6/12 CSI 1 2 l	Turns on (enables) local echo. When the terminal transmits characters, the characters are automatically sent to the terminal display.

4.6.6 Line Feed/New Line Mode (LNM)

Line feed/new line mode selects the control character(s) transmitted to the application by the RETURN and ENTER keys. ENTER transmits the same code as RETURN only when the auxiliary keypad is in Keypad numeric mode (DECKPNM).

Line feed/new line also selects the action taken by the terminal when receiving linefeed (LF), form feed (FF), or vertical tab (VT) codes. These three codes are always processed identically.

You set and reset linefeed/new line mode by using the following sequences.

NOTE: For compatibility with DIGITAL software, this mode should always be reset.

Mode	Sequence	Action
Set	9/11 3/2 3/Ø 6/8 CSI 2 Ø h	Causes a received LF, FF, or VT code to move the cursor to the first column of the next line. RETURN transmits both a CR and a LF code.
Reset	9/11 3/2 3/Ø 6/12 CSI 2 Ø l	Causes a received LF, FF, or VT code to move the cursor to the next line in the current column. RETURN transmits a CR code only.

4.6.7 Text Cursor Enable Mode (DECTCEM)

Text Cursor Enable Mode determines if the text cursor is visible.

You set and reset this mode using the following sequences.

Mode	Sequence	Action
Set	9/11 3/15 3/2 3/5 6/8 CSI ? 2 5 h	Causes the cursor to be visible.
Reset	9/11 3/15 3/2 3/5 6/12 CSI ? 2 5 l	Causes the cursor to not be visible.

4.6.8 Cursor Key Mode (DECCKM)

The characters generated by the cursor keys depend on the state of cursor key mode.

You set and reset this mode using the following sequences.

Mode	Sequence	Action
Set	9/11 3/15 3/1 6/8 CSI ? 1 h	Causes the cursor keys to generate "application" control functions.
Reset	9/11 3/15 3/1 6/12 CSI ? 1 l	Causes the cursor keys to generate ANSI cursor control sequences.

4.6.9 ANSI/VT52 Mode (DECANM)

In ANSI mode, reset selects VT52 compatibility mode. In VT52 mode, the terminal responds to private DIGITAL sequences like a VT52. The reset state of this mode sets the terminal to the VT52 mode. There is no Set state for this mode.

```
9/11  3/15  3/2  6/12
CSI    ?     2    l
```

Sets the terminal to VT52 mode.

4.6.10 Column Mode (DECCOLM)

Column mode selects the number of columns per line (80 or 132) on the display. The screen can display 24 lines of text with either selection. You select the number of columns per line by using the following sequences.

NOTE: When the terminal receives the sequence, the screen is erased and the cursor moves to the home position. This also sets the scrolling region for full screen (24 lines).

Mode	Sequence	Action
Set	9/11 3/15 3/3 6/8 CSI ? 3 h	Selects 132 columns per line.
Reset	9/11 3/15 3/3 6/12 CSI ? 3 l	Selects 80 columns per line.

4.6.11 Scrolling Mode (DECSCLM)

Scrolling is the upward or downward movement of existing lines on the screen. There are two methods of scrolling, jump scroll and smooth scroll (6 lines per second). You select the scroll mode (smooth or jump) using the following sequences.

NOTE: This is a User Preference feature and can be locked using Set-up.

Mode	Sequence	Action
Set	9/11 3/15 3/4 6/8 CSI ? 4 h	Selects smooth scroll. Smooth scroll lets the terminal add no more than 6 lines per second to the screen.
Reset	9/11 3/15 3/4 6/12 CSI ? 4 l	Selects jump scroll. Jump scroll lets the terminal add lines to screen as fast as possible.

4.6.12 Screen Mode (DECSCNM)

Screen mode selects either a dark or light (reverse) display background on the screen.

NOTE: This is a User Preference feature and can be "locked" by the terminal operator using Set-Up.

Mode	**Sequence**	**Action**
Set	9/11 3/15 3/5 6/8 CSI ? 5 h	Selects reverse video (dark characters on a light background).
Reset	9/11 3/15 3/5 6/12 CSI ? 5 l	Selects normal screen (light characters on a dark background).

4.6.13 Origin Mode (DECOM)

Origin mode allows cursor addressing relative to a user-defined origin. This mode resets when the terminal is powered up or reset. It does not affect the erase in display (ED) function.

Mode	**Sequence**	**Action**
Set	9/11 3/15 3/6 6/8 CSI ? 6 h	Select home position with line numbers starting at top margin of the user-defined scrolling region. The cursor cannot move out of the scrolling region.
Reset	9/11 3/15 3/6 6/12 CSI ? 6 l	Select home position in upper-left corner of screen. Line numbers are independent of scrolling region. Use the CUP sequence to move the cursor out of the scrolling region.

4.6.14 Auto Wrap Mode (DECAWM)

This mode selects where received graphic characters appear when the cursor is at the right margin.

NOTE: Regardless of this selection, the tab character never moves the cursor to the next line.

Mode	Sequence	Action
Set	9/11 3/15 3/7 6/8 CSI ? 7 h	Selects auto wrap. Graphic display characters received when the cursor is at the right margin appear on the next line. The display scrolls up if the cursor is at the end of the scrolling region.
Reset	9/11 3/15 3/7 6/12 CSI ? 7 l	Turns off auto wrap. Graphic display characters received when the cursor is at the right margin replace previously displayed characters.

4.6.15 Auto Repeat (DECARM)

Auto repeat mode selects automatic key repeating. When auto repeat mode is set, a key pressed for more than Ø.5 second automatically repeats the transmission of the character. The following keys never autorepeat: HOLD SCREEN, PRINT SCREEN, SET-UP, DATA/TALK, BREAK, RETURN, COMPOSE CHARACTER, LOCK, SHIFT, CTRL. You select auto repeat mode using the following sequences.

NOTE: This is a User Preference feature and can be "locked" by the opeator using Set-up.

Mode	Sequence	Action
Set	9/11 3/15 3/8 6/8 CSI ? 8 h	Selects auto repeat mode. A key pressed for more than Ø.5 second automatically repeats.
Reset	9/11 3/15 3/8 6/12 CSI ? 8 l	Turns off auto repeat. Keys pressed do not automatically repeat.

4.6.16 Print Form Feed Mode (DECPFF)

This mode determines if the terminal transmits a print termination character after a screen print. The form feed control character (FF) serves as the print termination character. You select print form feed mode using the following sequences.

Mode	Sequence	Action
Set	9/11 3/15 3/1 3/8 6/8 CSI ? 1 8 h	Selects form feed (FF) as print termination character. The terminal transmits this character to the printer after each print screen operation.
Reset	9/11 3/15 3/1 3/8 6/12 CSI ? 1 8 1	Select no termination character. The terminal does not transmit a form feed (FF) to the printer after each print screen operation.

4.6.17 Print Extent Mode (DECPEX)

Print extent mode selects the full screen or the scrolling region to print during a print screen operation. You select this mode using the following sequences.

Mode	Sequence	Action
Set	9/11 3/15 3/1 3/9 6/8 CSI ? 1 9 h	Selects full screen to print during a print screen operation.
Reset	9/11 3/15 3/1 3/9 6/12 CSI ? 1 9 1	Selects scrolling region to print during a print screen operation.

4.6.18 Keypad Mode (DECKPAM/DECPNM)

The auxiliary keypad generates either numeric characters or control functions. Selecting application or numeric keypad mode determines the type of characters.

NOTE: When the terminal is powered up or reset, the terminal selects numeric keypad mode.

Mode	Sequence	Action
Application (DECKPAM)	1/11 3/13 ESC =	Selects application keypad mode. Keypad generates "application" control functions.
Numeric (DECKPNM)	1/11 3/14 ESC >	Selects numeric keypad mode. Keypad generates characters that match the numeric, comma, period, and minus sign keys on main keypad. PF1 through PF4 generate control functions.

4.7 CURSOR POSITIONING

The cursor indicates the active screen position where the next character will appear in the absence of auto wrap. A number of operations implicitly affect cursor positioning. In addition, you can control cursor movement using the following sequences.

NOTE: Pn is a variable, ASCII coded, numeric parameter. If you select no parameter or a parameter value of Ø, the terminal assumes the parameter equals 1.

Name	Sequence	Action
Cursor Up (CUU)	9/11 4/1 CSI Pn A	Moves the cursor up Pn lines in the same column. The cursor stops at the top margin.
Cursor Down (CUD)	9/11 4/2 CSI Pn B	Moves the cursor down lines in the same column. The cursor stops at the bottom margin.
Cursor Forward (CUF)	9/11 4/3 CSI Pn C	Moves the cursor right columns. The cursor stops at the right margin.
Cursor Backward (CUB)	9/11 4/4 CSI Pn D	Moves the cursor left Pn columns. The cursor stops at the left margin.
Cursor Position (CUP)	9/11 3/11 4/8 CSI Pl ; Pc H	Moves the cursor to line Pl, column Pc. The numbering of the lines and columns depends on the state (set/reset) of origin mode (DECOM).

Horizontal And Vertical Position (HVP)	9/11 3/11 6/6 CSI Pl ; Pc f	Moves the cursor to line Pl, column Pc. The numbering of the lines and columns depends on the state (set/reset) of origin mode (DECOM). DIGITAL recommends using CUP instead of HVP.
Index (IND)	1/11 4/4 ESC D	IND is an 8-bit control character (8/4). It can be expressed as an escape sequence for a 7-bit environment. IND moves the cursor down one line in the same column. If the cursor is at the bottom margin the screen performs a scroll-up.
Reverse Index (RI)	1/11 4/13 ESC M	RI is an 8-bit control character (8/13). It can be expressed as an escape sequence for a 7-bit environment. RI moves the cursor up one line in the same column. If the cursor is at the top margin the screen performs a scroll-down.
Next Line (NEL)	1/11 4/5 ESC E	NEL is an 8-bit control character (8/5). It can be expressed as an escape sequence for a 7-bit environment. NEL moves the cursor to the first position on the next line. If the cursor is at the bottom margin the screen performs a scroll-up.
Save Cursor (DECSC)	1/11 3/7 ESC 7	Saves in terminal memory the: ● cursor position ● graphic rendition ● character set shift state ● state of wrap flag ● state of origin mode ● state of selective erase

Restore Cursor (DECRC)	1/11 3/8 ESC 8	Restores the states described for (DECSC) above. If none of these characteristics were saved: the cursor moves to home position, origin mode is reset, no character attributes are assigned, and the default character set mapping is established.

4.8 TAB STOPS

You select tab stop positions on the horizontal lines of the screen. The cursor advances (tabs) to the next tab stop when the terminal receives a horizontal tab code (HT, Ø/9). If there is no next tab, HT moves the cursor to the right margin. You set and clear the tab stops using the following sequences.

NOTE: These sequences are affected by the User Preference Lock in Set-up.

Name	Sequence	Action
Horizontal Tab Set (HTS)	1/11 4/8 ESC H	HTS sets a tab stop at the current column. HTS is an 8-bit control character (8/8) that you can also express as an escape sequence when coding for a 7-bit environment.
Tabulation Clear (TBC)	9/11 6/7 CSI g	Clears a horizontal tab stop at cursor position.
	9/11 3/Ø 6/7 CSI Ø g	Clears a horizontal tab stop at cursor position.
	9/11 3/3 6/7 CSI 3 g	Clears all horizontal tab stops.

4.9 CHARACTER RENDITION AND ATTRIBUTES

Character rendition and attributes are display features that affect the way a character is displayed without changing the character. You change character rendition using Select Graphic Rendition (SGR) sequences. You can also elect to designate characters to be selective erasable or not selective erasable when using erase sequences using the DECSCA sequence.

4.9.1 Select Graphic Rendition (SGR)

You can select one or more character renditions at a time using the following format:

```
9/11    3/11         6/13
CSI  Ps  ;   Ps ...   m
```

When you use multiple parameters, they are executed in sequence. The effects are cumulative. For example, to change from increased intensity to blinking-underlined, you can use:

```
9/11 3/Ø 3/11 3/4 3/11 3/5 6/13
CSI   Ø   ;    4   ;    5   m
```

When you select a single parameter, no delimiter (3/11) is used. For example, to select blinking only, you can use:

```
9/11 3/5 6/13
CSI   5   m
```

After you select an attribute, all new characters received by the terminal appear with that attribute. If you move the characters by scrolling, the attribute moves with the characters.

Select character attributes using the formats described above and the following Ps parameter values.

Ps	**Action**
3/Ø Ø	All attributes off
3/1 1	Display bold
3/4 4	Display underscored
3/5 5	Display blinking
3/7 7	Display negative (reverse) image
3/2 3/2 2 2	Display normal intensity
3/2 3/4 2 4	Display not underlined
3/2 3/5 2 5	Display not blinking
3/2 3/7 2 7	Display positive image

4.9.2 Select Character Attributes (DECSCA)

You can select all subsequent characters to be "selective erasable" or "not selective erasable" (see section on ERASING) using the following format:

NOTE: This sequence is supported only in VT200 mode.

```
9/11      2/2  7/1
CSI  Ps    "    q
```

where:

Ps	Action
0	All attributes off (does not apply to SGR)
1	Designate character as "non-erasable" by DECSEL/DECSED. (Attribute on)
2	Designate character "erasable" by DECSEL/DECSED. (Attribute off)

NOTE: A parameter value of 0 implies the default which is attributes off (erasable by DECSEL/DECSED). A parameter value of 2 is an explicit request for this attribute to be off (erasable by DECSEL/DECSED).

4.10 LINE ATTRIBUTES

Line attributes are display features that affect a complete display line. The cursor selects the line affected by the attribute. The cursor stays in the same character position when the attribute changes, unless the attribute would move the cursor past the right margin. In that case, the cursor stops at the right margin. When you move lines on the screen by scrolling, the attribute moves with the line. Select line attributes by using the following sequences.

NOTE: If you erase an entire line while using the erase in display (ED) sequence, the line attribute changes to single-height and single-width.

4.10.1 Double Height Line (DECDHL)

```
Top half          Bottom Half

1/11 2/3 3/3      1/11 2/3 3/4
ESC   #   3       ESC   #   4
```

These sequences make the line with the cursor the top or bottom half of a double-height, double width line. You must use these sequences in pairs on adjacent lines. The same character must be used on both lines to form full character. If the line was previously single-width, single-height, all characters to the right of center are lost.

4.10.2 Single-Width Line (DECSWL)

```
1/11 2/3 3/5
ESC   #   5
```

The DECSWL sequence makes the line with the cursor single-width, single-height. This is the line attribute for all new lines on the screen.

4.10.3 Double-Width Line (DECDWL)

```
1/11 2/3 3/6
ESC   #   6
```

The DECDWL sequence makes the line with the cursor double-width, single-height. If the line was previously single-width, single-height, all characters to the right of center screen are lost.

4.11 EDITING

You use editing sequences to insert and delete characters and lines of characters at the cursor position. The cursor position does not change when inserting or deleting lines. Delete characters or insert and delete lines by using the following sequences.

NOTE: Pn is a variable, ASCII coded, numeric parameter. If you select no paramater or a parameter value of 0, the terminal assumes a parameter value of 1.

Name	Sequence	Action
Insert Line (IL)	9/11 4/12 CSI Pn L	Inserts Pn lines at the cursor. If fewer than Pn lines remain from the current line to the end of the scroll region, the number of lines inserted is the lesser number. Lines within the scroll region at and below the cursor move down. Lines moved past the bottom margin are lost. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region.

Name	Sequence	Action
Delete Line (DL)	9/11 4/13 CSI Pn M	Deletes Pn lines starting at the line with the cursor. If fewer than Pn lines remain from the current line to the end of the scroll region, the number of lines deleted is the lesser number. As lines are deleted, lines within the scroll region and below the cursor move up, and blank lines are added at the bottom of the scroll region. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region.
Insert Characters (ICH) (VT2ØØ mode only)	9/11 4/Ø CSI Pn @	Insert Pn blank characters at the cursor position, with the character attributes set to normal. The cursor does not move and remains at the beginning of the inserted blank characters. A parameter of Ø or 1 causes one blank character to be inserted. Data on the line is shifted forward as in character insertion.

Name	Sequence	Action
Delete Character (DCH)	9/11 5/0 CSI Pn P	Deletes Pn characters starting with the character at the cursor position. When a character is deleted, all characters to the right of the cursor move to the left. This creates a space character at the right margin for each character deleted. Character attributes move with the characters. The spaces created at the end of the line have all their character attributes off.

4.12 ERASING

Erasing removes characters from the screen without affecting other characters on the screen. Erased characters are lost. The cursor position does not change when erasing characters or lines.

Erasing a character also erases any character attribute of the character. You erase characters by using the following sequences.

Name	Sequence	Action
Erase Character (ECH) (VT200 mode only)	9/11 5/8 CSI Pn X	Erases characters at the cursor position and the next n-1 character. A parameter of 0 or 1 causes a single character to be erased. Character attributes are set to normal. No reformatting of data on the line occurs. The cursor remains in the same position.

Name	Sequence	Action
Erase in Line (EL)	9/11 4/11 CSI K	Erases from the cursor to the end of the line, including the cursor position. Line attribute is not affected.
	9/11 3/Ø 4/11 CSI Ø K	Same as above.
	9/11 3/1 4/11 CSI 1 K	Erases from the beginning of the line to the cursor, including the cursor position. Line attribute is not affected.
	9/11 3/2 4/11 CSI 2 K	Erases the complete line.
Erase in Display (ED)	9/11 4/1Ø CSI J	Erases from the cursor to the end of the screen, including the cursor position. Line attribute becomes single-height, single-width for all completely erased lines.
	9/11 3/Ø 4/1Ø CSI Ø J	Same as above.
	9/11 3/1 4/1Ø CSI 1 J	Erases from the beginning of the screen to the cursor, including the cursor position. Line attribute becomes single-height, single width for all completely erased lines.
	9/11 3/2 4/1Ø CSI 2 J	Erases the complete display. All lines are erased and changed to single-width. The cursor does not move.

Name	Sequence	Action
Selective Erase In Line (DECSEL) (VT2ØØ mode only)	9/11 3/15 4/11 CSI ? K	Erases all "erasable" characters (DECSCA) from the cursor to the end of the line. Does not affect video line attributes or video character attributes (SGR).
	9/11 3/15 3/Ø 4/11 CSI ? Ø K	Same as above.
	9/11 3/15 3/1 4/11 CSI ? 1 K	Erases all "erasable" characters (DECSCA) from the beginning of the line to and including the cursor position. Does not affect video line attributes or video character attributes.
	9/11 3/15 3/2 4/11 CSI ? 2 K	Erases all "erasable" characters (DECSCA) on the line. Does not affect video line attributes or video character attributes.
Selective Erase In Display (DECSED) (VT2ØØ mode only)	9/11 3/15 4/1Ø CSI ? J	Erases all "erasable" characters (DECSCA) from and including the cursor end of the screen. Does not affect video line attributes or video character attributes (SGR).
	9/11 3/15 3/Ø 4/1Ø CSI ? Ø J	Same as above.
	9/11 3/15 3/1 4/1Ø CSI ? 1 J	Erases all "erasable" characters (DECSCA) from the beginning of the screen to and including the cursor. Does not affect video line attributes or video character attributes (SGR).

Name	Sequence	Action
	9/11 3/15 3/2 4/1Ø CSI ? 2 J	Erases all "erasable" characters (DECSCA) in the entire display. Does not affect video character attributes or video line attributes (SGR).

4.13 SCROLLING MARGINS (TOP AND BOTTOM)

The scrolling region is the area of the screen that can receive new characters by scrolling old characters off the screen. The area is defined by the top and bottom screen margins. The minimum size of the scrolling region allowed is two lines; therefore the number of the top margin must be at least one less than the number of the bottom margin. You select the top and bottom margins of the scrolling region by using the following sequence.

4.13.1 Set Top and Bottom Margins (DECSTBM)

```
9/11      3/11      7/2
CSI  Pt   ;    Pb    r
```

This sequence selects top and bottom margins defining the scrolling region. Pt is the line number of the first line in the scrolling region. Pb is the line number of the bottom line. If either Pt or Pb is not selected, they default to top and bottom respectively. Lines are counted from "1".

NOTE: Power up or reset causes the scrolling region to be the complete screen.

4.14 PRINTING

All print operations are selectable via control sequences. When characters are printed on the screen, terminal and printer tab stops are ignored. Print characters are spaced with the space (SP) character. The terminal transmits a carriage return (CR) and line feed (LF), or vertical tab (VT) or form feed (FF) after the last printable character of a line (not a space character).

NOTE: Spaces with video attributes are "printable" characters.

Before you select a print operation, you should check printer status using the print status report (DSR) (see Reports section). You select print operations using the following sequences.

Name	Sequence	Action
Auto Print Mode	9/11 3/15 3/5 6/9 CSI ? 5 i	Turns on auto print mode. Subsequent display lines print when you move the cursor off the line using a linefeed, form feed, vertical tab, or autowrap. The printed line is terminated with a carriage return and the character which moved the cursor off the previous line (LF, FF, or VT (autowrap lines end with a linefeed).
	9/11 3/15 3/4 6/9 CSI ? 4 i	Turns off auto print mode.
Printer Controller	9/11 3/5 6/9 CSI 5 i	Turns on printer controller mode. The terminal transmits received characters to the printer without displaying them on the screen. All characters and character sequences except NUL, XON, XOFF, CSI 5 i, and CSI 4 i are sent to the printer. The terminal does not insert or delete spaces, or provide line delimiters, or select the correct printer character set. Printer Controller mode is of higher priority than Auto Print mode. It can be selected during Auto Print mode. When in Printer Controller mode, keyboard activity continues to be directed to the host.
	9/11 3/4 6/9 CSI 4 i	Turns off printer controller mode.

Name	Sequence	Action
Print Cursor Line	9/11 3/15 3/1 6/9 CSI ? 1 i	Prints the display line containing the cursor. The cursor position does not change. The print-cursor-line sequence is completed when the line prints.
Print Screen	9/11 6/9 CSI i	Prints the screen display (full screen or scrolling region, depending on the Print Extent DECEXT selection). Printer form feed mode (DECPFF) selects either a form feed (FF) or nothing as the print terminator. The print screen sequence is completed when the screen prints.
	9/11 3/Ø 6/9 CSI Ø i	Same as above.

4.15 USER DEFINED KEYS (DECUDK)

Fifteen of the terminal's top-row function keys are programmable: F6 through F14, DO, HELP, and F17 through F2Ø (HOLD SCREEN, PRINT SCREEN, SET-UP, DATA/TALK and BREAK have dedicated local functions and are not programmable). When the terminal is in VT2ØØ mode, you can down-line load one or more key sequences for the programable function keys by using DECUDK device control strings (the programmable function keys are inoperative in the VT1ØØ and VT52 modes).

The programmed values of the keys are accessed by typing SHIFT-(function key), while the normal control sequence values are accessed by typing the function key alone.

There are 256 bytes available to the 15 programmable function keys. Space is supplied on a first-come/first-serve basis. Once the 256 bytes are used, no more keys can be redefined unless you free up space. You can free up space either by redefining a key or keys using a DECUDK, clearing a key or keys using a DECUDK, or clearing the definition set through a terminal power-up or Recall function.

NOTE: All key definitions are stored in volatile RAM. Loss of terminal power results in loss of UDK key definitions.

4.15.1 DECUDK Device Control String Format

NOTE: Refer to Chapter 2 for general information about device control strings.

The device control string format for down-line loading UDK functions is:

DCS	Pc;Pl	|	Ky1/st1;ky2/st2;...kyn/stn	ST
Control String Introducer	Clear and Lock Parameters	Final Character	Key Definition String	String Terminator

Each string component is described below.

NOTE: This sequence is supported only in VT200 mode.

The device control string introducer is DCS (9/0). This character introduces the control string. DCS is an 8-bit character (9/0) that you can also express as ESC P (1/11 5/0) when coding for a 7-bit environment.

The clear parameter (Pc) determines which keys are cleared and when. A value of zero ("clear all") clears all keys then loads each specific key as it is encountered in the DRCS. A value of one ("load new values, clear old only when redefined") clears each key that is to be reloaded just before loading it and does not clear keys which are not being redefined. By using a value of 1 for Pc you can redefine some keys without reloading them all.

NOTE: There are only 256 bytes available. Any particular key can be defined to contain no more than 256 bytes or the number of bytes available when that key is loaded, whichever is less.

Note that if you set the clear parameter to 1 (load new, but do not clear old) it is possible that a key load might fail because of lack of room even though the final total for all keys would have been 256 bytes or less. The reason for this is as follows: With Pc value set to 1, keys are cleared and loaded sequentially. Sequential loading could result in intermediate storage requirements higher than 256 bytes, even though the final requirement would 256 bytes or less. For example: if F6 contained 120 bytes, F7 contained 110 bytes, and F8 contained 20 bytes, loading F8 with 40 bytes, F6 with 1 byte and F7 with 1 byte works if all keys are cleared first but not if the keys are cleared as they are sequentially redefined. When you attempt to load F8 with 40 bytes, the load fails because only 26 bytes are free at that time (256 - 120 - 110 = 26).

The following is a summary of Pc values and meanings:

Pc	Meaning
none	Clear all keys before loading new values
Ø	Clear all keys before loading new values
1	Load new key values, clear old only where defined

The lock parameter (Pl) determines whether the key definitions are locked or not locked after you load it. It follows the Pc values and is separated by a semicolon character (; 3/11) as a delimiter. If you set the Pl value to Ø (lock) the keys are locked at the completion of loading. At this point the terminal operator must unlock the keys for redefinition using Set-up. If you set the Pl value to 1 (do not lock), the keys are available for further definition with another DECUDK string. The default for the lock parameter is lock.

NOTE: A Pl value of 1, does not unlock the keys, it simply does not lock them.

The following is a summary of Pl values and meanings:

Pl	Meaning
None	Lock the keys against future redefinition
Ø	Lock the keys against future redefinition
1	Do not lock the keys against future redefinition

The final character, a vertical bar (7/12) designates this control string as a DECUDK.

The key definition strings (Kyn/Stn) are included in the data following the final character and before the string terminator. Each key definition string consists of a key selector number (Kyn) and a string parameter (Stn) separated by a slash (/, 2/15). The key selector numbers (Kyn) specify the particular key to be redfined, and the string parameters (Stn) are the encoded contents of the keys. The string parameters (Stn) consist of hex pairs in the range of 3/Ø through 3/9 ("Ø" through "9"), 4/1 through 4/6 ("A" through "F"), and 6/1 through 6/6 ("a" through "f"). When you combine these hex values they represent an 8-bit quantity. This allows any of the 256 character codes to be used in the key sequence. You can use key definition strings in any order and can specify multiple definitions using a semicolon (; 3/11) as a delimiter.

The following is a list of definable keys and their identifying values:

Key	Value
F6	17
F7	18
F8	19
F9	20
F10	21
F11	23
F12	24
F13	25
F14	26
HELP	28
DO	29
F17	31
F18	32
F19	33
F20	34

The string terminator is ST (9/12). This is an 8-bit control character that you can also express as ESC \ (1/11 5/12) when coding for a 7-bit environment.

4.15.2 Things to Keep in Mind When Loading Keys

The following is a list of general information you should keep in mind when loading the keys:

- Software should use the UDK function to reclaim key definition space. You can do this by clearing keys without locking them. Once the keys have been cleared, you can use the UDK function to redefine the keys and lock them.

- Generally, you should not leave keys unlocked. This could cause a breach of security for the terminal user and the computer system.

- The host must keep track of space available for definitions.

- If you redefine a key the old sequence is lost. This may free up space if the new sequence is shorter than the previous definition.

- The terminal uses a special lock to arbitrate the programming of keys. This lock can be turned on or off through Set-up. It may also be turned on with a DECUDK from the host. The lock acts globally over all programmable keys.

- The default value for each key is empty (blank). When you clear the keys they are set back to the empty condition. All key definitions are stored in volatile RAM. Therefore a loss of power to the terminal results in the loss of all key definitions. An aborted function key load (by error or other means) locks the keys, saves the already successfully loaded fraction, and sends the rest of the DECUDK sequence to the screen. An invalid DCS hex pair in a key definition string results in an aborted load.

4.15.3 Examples and Recommendations for Using DECUDK

To clear keys send the following string:

```
9/Ø  3/Ø  3/11  3/1  7/12  9/12
DCS   Ø    ;     1    |     ST
```

To lock keys send the following string:

```
9/Ø  3/1  3/11  3/Ø  7/12  9/12
DCS   1    ;     Ø    |     ST
```

Suppose you want to define key F2Ø to be "PRINT", and you want to do this without clearing or locking any other keys. The first part of the string would be:

```
9/Ø  3/1  3/11  3/1  7/12  3/3  3/4  2/15
DCS   1    ;     1    |     3    4    /
```

The 34 after the final character (7/12) identifies key F2Ø. After the slash character (2/15), you would provide the definition. The encoding for PRINT is:

P = 5Ø hex
R = 52 hex
I = 49 hex
N = 4E hex
T = 54 hex

Thus, after the slash character, you would provide:

```
3/5  3/Ø  3/5  3/2  3/4  3/9  3/4  4/5  3/5  3/4  9/12
 5    Ø    5    2    4    9    4    E    5    4   ST
```

The ST character (9/12) specifies the end of the string.

4.16 DOWN-LINE LOADABLE CHARACTER SET

The VT22Ø, when in a VT2ØØ mode, lets you create and down-line load a character set containing up to 94 characters. This character set is called a Dynamically Redefinable Character Set (DRCS). After you have created characters, you can load them into the terminal DRCS buffer using a DECDLD device control string.

NOTE: This character set is not loaded into non-volatile RAM. Therefore, when the terminal is powered off, characters are lost.

4.16.1 Designing a Character Set

Figure 4-4 shows the relationship between DRCS cell size and the terminals normal character cell size in pixels. Note that the maximum number of pixels in the DRCS cell is 8Ø (8 X 1Ø). Note also, that the normal terminal character cell size is less than the DRCS cell size. Since the terminal ignores characters defined beyond the DRCS cell size, you must design your characters to fit the DRCS cell size.

Each pixel in a character font is represented by a bit with a binary value of 1 (on) or Ø (off). One (1) specifies foreground (pixel on) and zero (Ø) specifies background (pixel off).

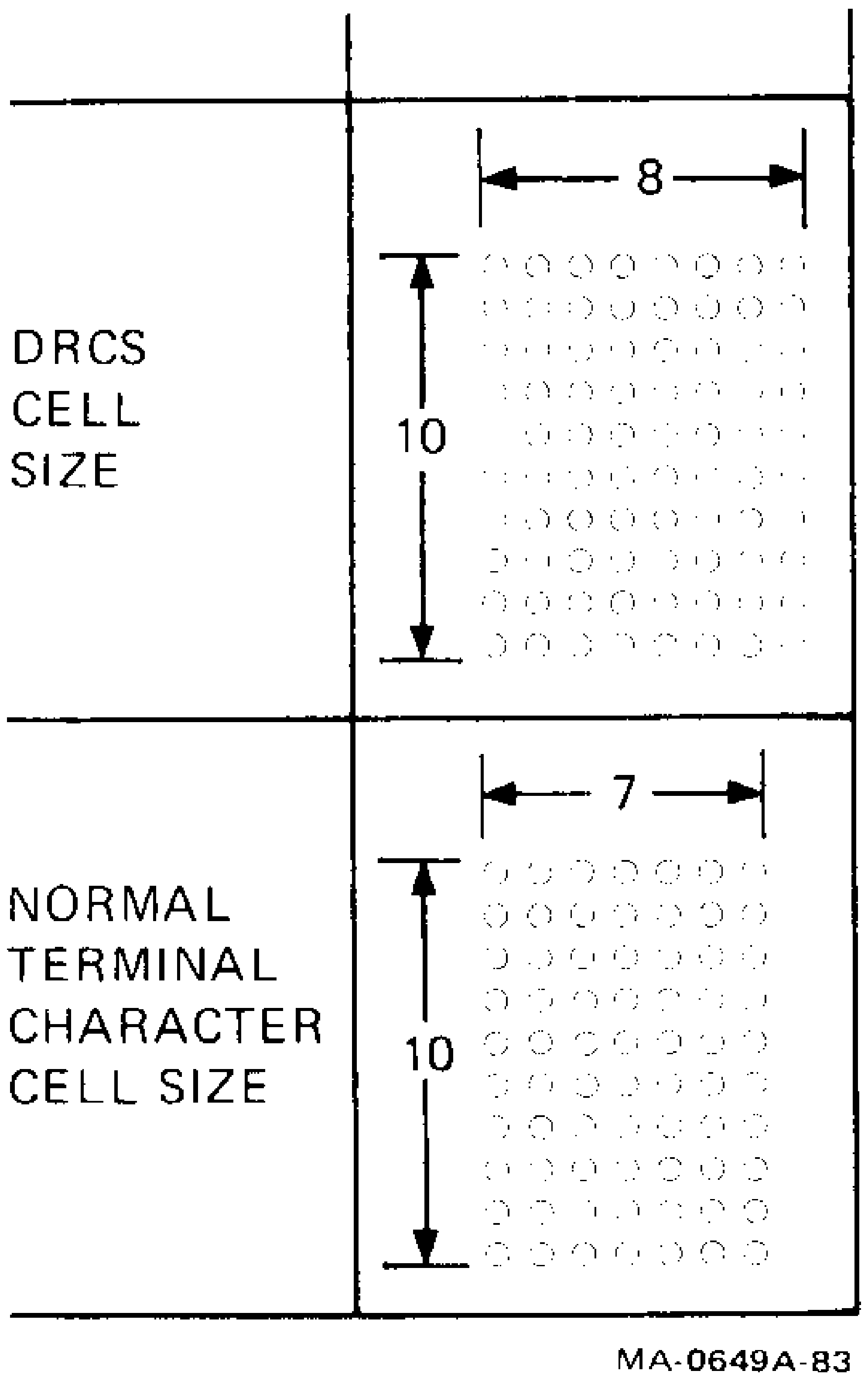


Figure 4-4 Cell and Normal Character Cell Size Relationship

For example, suppose you want to design character A. To do this, you designate which pixels are to be on and which pixels are to be off. Your design may look like what is shown in Figure 4-5.

After you have established what your DRCS character A is going to look like, you then divide the pixels of the DRCS character cell into columns of six bits each using the format shown in Figure 4-6. The column numbers here designate the order in which the columns will be sent to the terminal. Each column is now represented as a vertical 1 X 6 pixel matrix called a "sixel" with the least significant bit at the top and the most significant bit at the bottom. Because the character height (1Ø pixels) is not a multiple of six, the columns on the bottom the character cell consist of only four bits each (the two highest order bits, 5 and 6, are ignored).

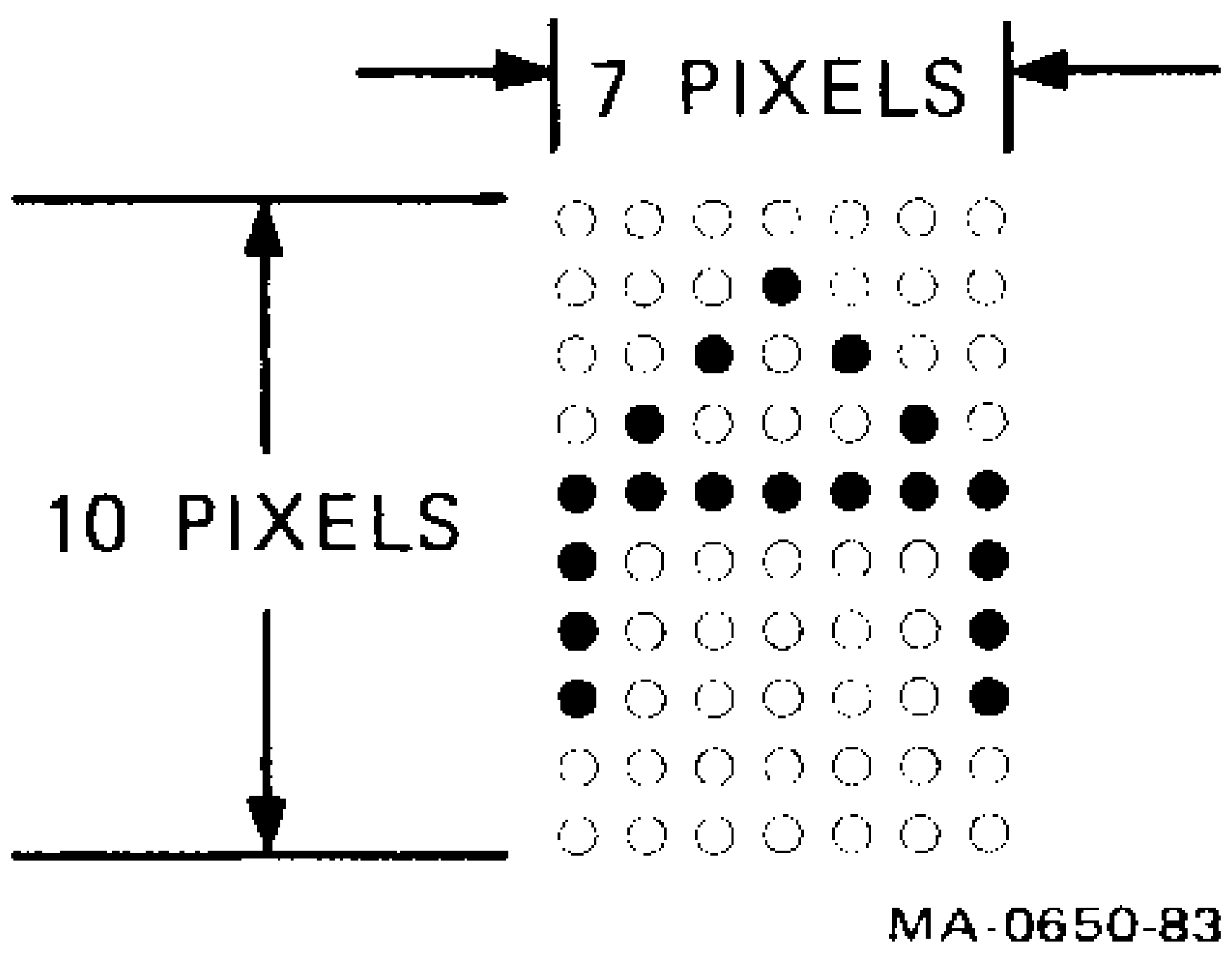


Figure 4-5 Example of an "A" Character

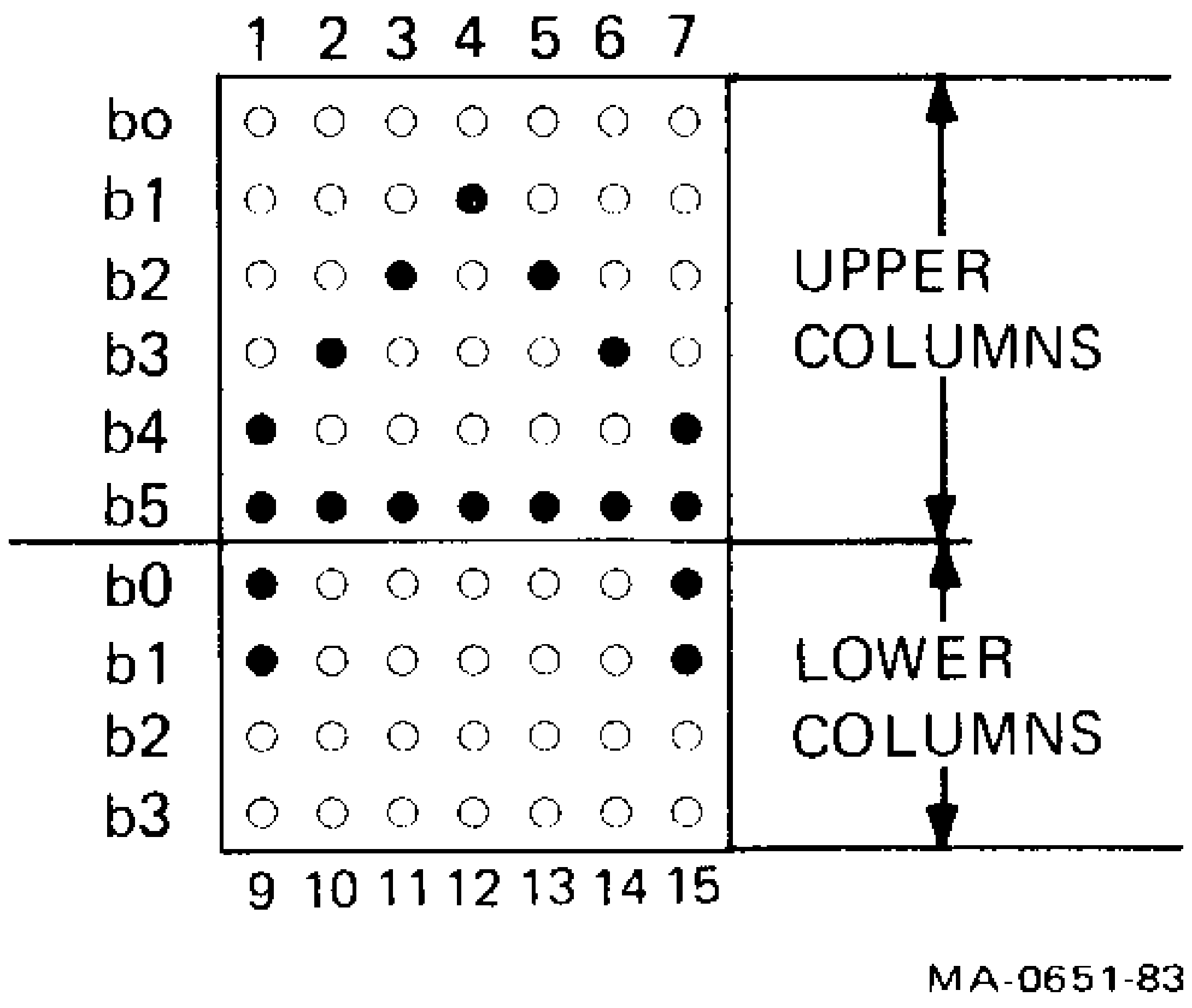


Figure 4-6 Example of "A" Divided Into Columns (6 Bits per Column)

After you have divided your DRCS character into six-pixel columns ("sixels"), you then convert the binary values of each column to its equivalent character. Because column codes are restricted to characters with the range of ? (octal Ø77) to ~ (octal 176), you must add an offset of octal Ø77 to each column octal value. Thus, binary value ØØØØØØ is converted to octal Ø77 (octal Ø + octal 77); binary 11Ø1Ø1 is converted to octal 164 (octal 65 + octal Ø77); and binary 111111 is converted to octal 176 (octal Ø77 + Ø77).

After you have converted the binary column codes to octal values (using the offset), you then convert the resultant octal value for each column to its equivalent character using the ASCII table in Chapter 2. Figure 4-7 provides this conversion procedure for our example DRCS character A.

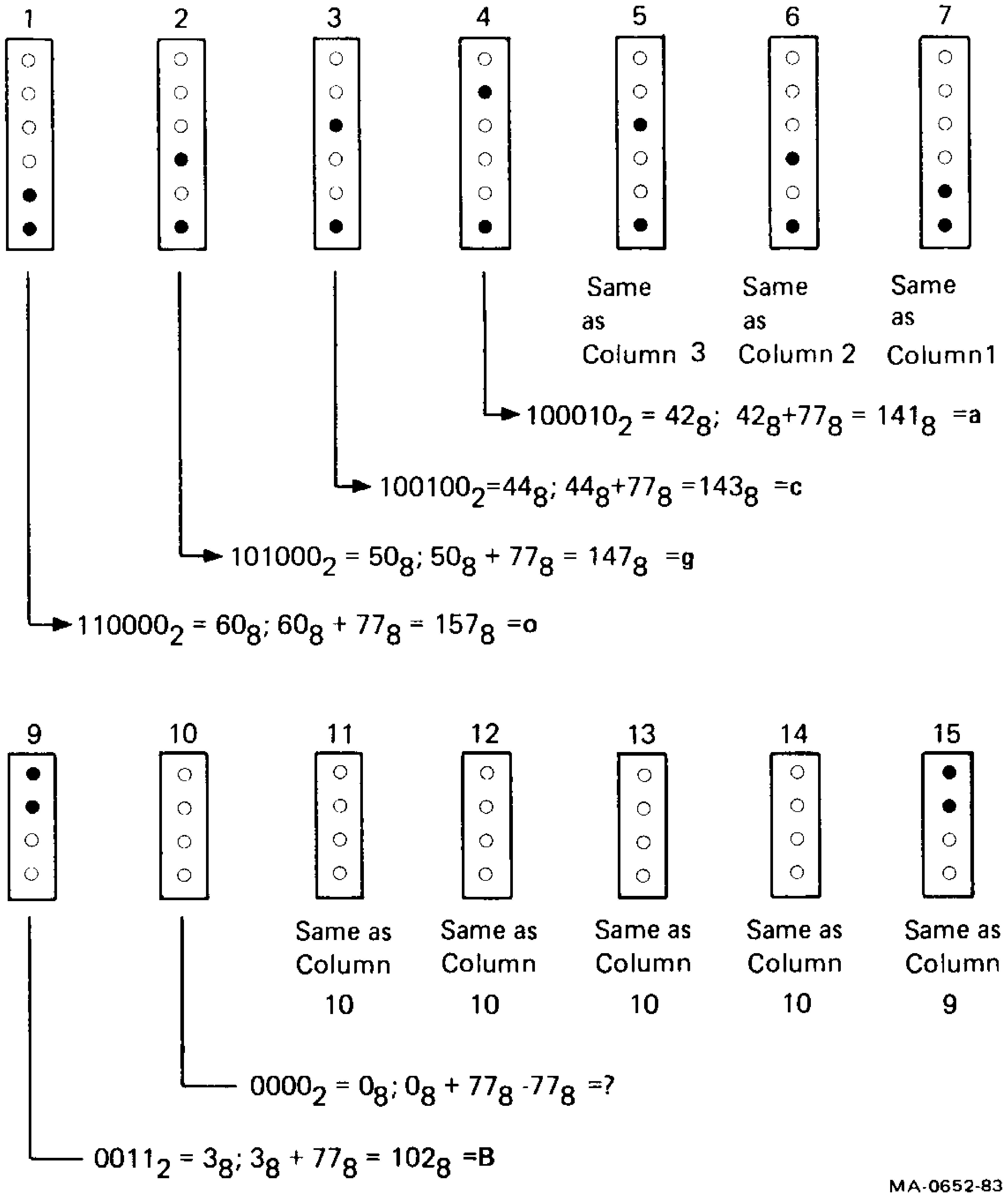

1 2 3 4 5 6 7
Same as Column 3
Same as Column 2
Same as Column1
$100010_2 = 42_8$; $42_8 + 77_8 = 141_8$ =a
$100100_2 = 44_8$; $44_8 + 77_8 = 143_8$ =c
$101000_2 = 50_8$; $50_8 + 77_8 = 147_8$ =g
$110000_2 = 60_8$; $60_8 + 77_8 = 157_8$ =o
9 10 11 12 13 14 15
Same as Column 10
Same as Column 10
Same as Column 10
Same as Column 10
Same as Column 9
$0000_2 = 0_8$; $0_8 + 77_8 \text{-} 77_8$ =?
$0011_2 = 3_8$; $3_8 + 77_8 = 102_8$ =B
MA-0652-83

Figure 4-7 Column Codes for Example 8Ø-Column Font Character A

After you have designed your DRCS characters using the conversion procedure described, you can then down-line load your DRCS characters which consist of a string or strings of characters using the DECDLD device control string described in the next subparagraph.

4.16.2 Down-Line Loading DRCS Characters

You can down-line load your DRCS character set using the following DECDLD device control string format:

NOTE: Refer to Chapter 2 for general information about device control strings.

DCS Pfn;Pcn;Pe;Pcms;Pw;Pt { Dscs Sxbpl;Sxbp2;...;Sxbpn ST

Where:

DCS (9/Ø) is the device control string introducer. It is an 8-bit control character that you can also express as ESC P (1/11 5/Ø) when coding for a 7-bit environment.

Pfn;Pcn;Pe;Pcms;Pw;Pt are parameter characters, separated by semicolons, that define parameters described in Table 4-9. Valid values for Pcms, Pw, and Pt are: Pcms = Ø, 2, 3 or 4; Pw = Ø, 1 or 2; and Pt = Ø or 1. Invalid combinations are ignored. A font loaded for 8Ø columns can be used in 132 columns and a font loaded for 132 columns can be used in 8Ø columns.

{ (7/11) is the final character that signals the end of the parameter characters and specifies a DECDLD function.

Dscs defines the character set "name" for the soft font, and is used in the SCS (select character set) escape sequence.

Sxbpl;Sxbp2;...;Sxbpn are sixel bit patterns (1 to 94 patterns) for characters separated by semicolons. Each sixel bit pattern has the form:

S...S/...S

where the first S....S represents the upper columns (sixels) of the DRCS character, the slash (2/5) advances the sixel pattern to the lower columns of the DRCS character, and the second S....S represents the lower columns (sixels) of the DRCS (see Figure 4-6).

ST (9/12) is the string terminator. it is an 8-bit control character that you can also express as ESC \ (1/11 5/12) when coding for a 7-bit environment.

Table 4-9 DECDLD Parameter Characters

Parameter	Name	Description
Pfn	Font Number	Specifies DRCS font buffer to be loaded. The VT22Ø has only one DRCS font buffer. This parameter has two valid values: Ø and 1.
Pcn	Starting Character Number	Selects starting character in DRCS font buffer to be loaded. For example: Parameter value 1 specifies a column 2/row 1 character, parameter 94 specifies a column 7/row 14 character (Table 2-1 in Chapter 2).
Pe	Erase Control	Selects which characters are erased before loading according to: Ø = erase all characters in this DRCS set 1 = erase only the characters that are being reloaded 2 = erase all characters in all DRCS sets (this font buffer number and other font buffer numbers)
Pcms	Character Matrix Size	Defines the expected limit of the character matrix size according to: Ø = Device default (7 X 1Ø) 1 = (not used) 2 = 5 X 1Ø 3 = 6 X 1Ø 4 = 7 X 1Ø
Pw	Width Attribute	Specifies the width attribute according to: Ø = Device default (8Ø Columns) 1 = 8Ø column 2 = 132 column
Pt	Text/ Full-Cell	Allows software to treat the font as a text font or a full-cell font according to: Ø = Device default (text) 1 = Text 2 = Full-Cell (not used) Full-Cell fonts can individually address all pixels in a cell, while text fonts, in general, may not be able to address all pixels individually.

4.16.3 DECDLD Example

Suppose you want to load a character set with the example character A designed at the begining of this paragraph as the first character. To do this, you could use the following device control string:

```
DCS 1;1;1 { ogcacgo/B?????B;(next character);.....ST
```

DCS introduces the sequence.

1;1;1 specifies loading the DRCS font buffer, selects starting character as column 2/row 1 of ASCII chart (Chapter 2), selects to erase only the character that are loaded (refer to Table 4-9). Note that Pcms, Pw, and Pt are not included; they default to zero values.

{ indicates the end of the parameter characters and specifies that this is a DECDLD control string.

ogcacgo are the character codes for the upper columns of the example DRCS character A.

/ advances the sixel sequence to the lower columns of the example DRCS character A.

B?????B are the character codes for the lower columns of the example DRCS character A.

; signals the end of the DRCS character being loaded and signals the beginning of another DRCS character to be loaded.

ST indicates the end of the device control string.

4.16.4 Clearing a Down-Line Loaded Character Set

You can clear a character set that you have down-line loaded using the following DECDLD control sequence:

```
    DCS 1;1;2 { sp @ ST
```

Down-line loaded character sets are also cleared by:

- performing the power-up self test
- using the Set-Up Recall or Default functions
- using RIS or ESC c sequences

4.17 REPORTS

The terminal, in response to requests from the host computer, transmits reports to provide identification (type of terminal), cursor position, and terminal operating status. There are two categories of reports: Device Attributes (DA) and Device Status Reports (DSR).

4.17.1 Device Attributes (DA)

There are two DA exchanges (dialogues) between the host computer and the VT22Ø: Primary DA and Secondary DA.

In the Primary DA exchange (the first exchange), the host asks for the terminal's service class code and the basic attributes. In the Secondary DA exchange (the second exchange), the host asks for the terminal's identification code, firmware version level, and an account of the hardware options installed. A typical DA exchange is described as follows:

Communication	Sequence	Meaning
Host to VT22Ø (Primary DA Request)	CSI c or CSI Ø c	"What is your service class code and what are your attributes?"
VT22Ø to Host (Primary DA Request)	CSI ? 62; 1; 2; 6; 7; 8 c	"I am a service class 2 (VT2ØØ family) terminal (62) with 132 columns (1), printer port (2), selective erase (6), DRCS (7), and UDK (8).
Host to VT22Ø (Secondary DA Response)	CSI > c or CSI > Ø c	"What type of terminal are you what is your firmware version, and what hardware options do you have installed?"
VT22Ø to Host (Secondary DA Response)	CSI > 1; Pv; Po c	"I am a VT22Ø (identification code of 1,) my firmware version is ____ (Pv), and I have PØ options installed.

EXAMPLE: CSI>1;1Ø;Øc = VT22Ø version 1.Ø, no options

4.17.2 Device Status Report (DSR)

In a DSR exchange, the host computer asks for general operating status of the terminal and/or printer. If the terminal is in Printer Controller mode, the printer will receive the DSR request but will not be able to answer.

DSR -- VT220

A typical DSR exchange is described as follows:

Communication	Sequence	Meaning
Host to VT220 (Request for terminal status)	CSI 5 n	"Please report your operating status using a DSR control sequence. Are you in good operating condition or do you have a malfunction?"
VT220 to Host (DA response)	CSI Ø n	"I have no malfunction."
	or	
	CSI 3 n	"I have a malfunction."
Host to VT220 (Request for cursor position)	CSI 6 n	"Please report your cursor position using a CPR (not DSR) control sequence."

VT220 to Host (CPR response)	CSI Pv; Ph R	"My cursor is positioned at ____ (Pv); ____ (Ph)."
		Where:
		Pv = vertical position (row)
		Ph = horizontal position (column)

DSR -- Printer Port

NOTE: Printer status should be determined before entering any print mode or starting any print function.

A typical DSR exchange is described as follows:

Communication	Sequence	Meaning
Host to VT220 (request for printer status)	CSI ? 15 n	"What is the printer status?"
VT220 to Host	CSI ? 13 n	"DTR has not been asserted on the printer port since power up or reset -- in essence, I have no printer."
	CSI ? 10 n	"DTR is asserted on the printer port. The printer is ready."
	CSI ? 11 n	"DTR is not currently asserted on the printer port. The printer is not ready."

DSR -- User Defined Keys (VT200 mode only)

Host to VT220 (request for UDK status)	CSI ? 25 n	"Are User Defined Keys locked or unlocked?"
VT220 to Host	CSI ? 20 n	"User Defined Keys are unlocked."
	CSI ? 21 n	"User Defined Keys are locked."
	CSI ? 21 n	"User Defined Keys are locked."

4.17.3 Identification (DECID)

The DECID sequence causes the terminal to send a primary DA response sequence. DECID, however, is not recommended. You should use the primary DA request for this purpose.

The DECID sequence is:

```
1/11 5/10
ESC   Z
```

4.18 TERMINAL RESET (DECSTR and RIS)

There are two terminal reset escape sequences. One causes a "soft" terminal reset (DECSTR), and the other causes a "hard" terminal reset (RIS).

4.18.1 Soft Terminal Reset (DECSTR)

DECSTR (soft terminal reset) can be invoked anytime by the terminal operator using RESET TERMINAL in Set-Up. It can be invoked directly from the host computer via the DECSTR sequence only when the terminal is in a VT2ØØ mode (when the terminal is in VT1ØØ or VT52 mode, the escape sequence is ignored). It can also be invoked indirectly via the DECSCL sequence (ignored in VT52 mode).

The DECSTR sequence sets the terminal to the power-up default states listed in Table 4-1Ø.

The DECSTR escape sequence is:

```
9/11 2/1 7/Ø
CSI   !   p
```

4.18.2 Hard Terminal Reset (RIS)

RIS (hard terminal reset or reset-to-initial state) can be invoked anytime by the terminal operator using RECALL in Set-Up. It can also be invoked from the host computer anytime via an escape sequence. RIS causes an NVR recall. All Set-Up parameters are replaced by their NVR values, or power-up default values if NVR values do not exist.

In addition, RIS:

- performs a communications line disconnect
- clears UDKs
- clears a down-line loaded character set
- clears the screen
- returns the cursor to the upper-left corner or the screen
- sets the SGR state to normal
- sets the selective erase attribute write state to "non-selective erasable"
- sets all character sets to the default

The RIS escape sequence is:

```
1/11 6/3
ESC   c
```

NOTE: This sequence should be used with caution. Parity and baud rate are restored from NVR.

Table 4-10 Soft Terminal Reset (DECSTR) States

Sequence	State	Stored in NVR
Text Cursor	On	Yes, NVR value ignored
Insert/Replace	Replace	No
Origin Mode	Absolute	No
Autowrap	Off	Yes, NVR value ignored
Keyboard Action	Unlocked	No
Keypad Mode	Numeric	No
Cursor Key Mode	Normal	No
Top Margin	1	No
Bottom Margin	24	No
Character Sets G0, G1, G2, G3 GL, GR	VT200 default when in VT200 mode or, via Set-Up only, to VT100 defaults when in VT52 or VT100 mode.	No
Video Character Attributes	Normal	No
Selective Erase Attributes	Normal (erasable by DECSEL/DECSED)	No
*Save Cursor State		No
Cursor Position	Home	
Character Sets	VT100 or VT200 defaults (as appropriate)	
Selective Erase Attribute Bit Write State	Off	
SGR Write State	Normal	
Origin Mode	Normal (reset)	
Character Shift (G0 to GL, G2 to GR no shifts)	Power Up Defaults	

* Applies only to subsequent restore cursor commands (DECRC)

4.19 TESTS AND ADJUSTMENTS (DECTST and DECALN)

The terminal has tests and alignment patterns that can be invoked from the keyboard or the host computer via control and escape sequences. Test and alignment procedures are usually performed only by DIGITAL manufacturing and Field Service personnel.

This section provides the sequences used to invoke the tests and the alignment patterns. For detailed information, however, refer to the VT220 Pocket Service Guide.

4.19.1 Tests (DECTST)

The sequence format for invoking terminal tests is:

```
9/11 3/4  3/11     3/11    3/11       7/9
CSI   4    ;    Ps    ; ..... ;        Ps y
```

Each Ps is the parameter indicating a test to be performed. After the first parameter (4), the parameters each select one test from the following list. Several tests can be invoked at one time by chaining the parameters together, separated by semicolons. The test are not necessarily executed in the order in which they are entered.

NOTE: DECTST causes a communications line disconnect.

Parameter	Test
0	Test 1, 2 3, and 6
1	Power-Up Self Test
2	EIA Port Data Loopback Test
3	Printer Port Loopback Test
4	(not used)
5	(not used)
6	EIA Port Modem Control Line Loopback Test
7	20 mA Port Loopback Test
8	(not used)
9	Repeat any selected test continuously until power off or failure
10 and up	(not used)

4.19.2 Adustments (DECALN)

The terminal has a screen alignment pattern that service personnel use to adjust the screen. You can display the screen alignment pattern using the DECALN sequence.

```
1/11 2/3 3/8
ESC   #   8
```

This sequence fills the screen with upper case E's.

4.2Ø VT52 MODE ESCAPE SEQUENCES

The VT52 mode allows the VT22Ø to operate with DIGITAL software written for the VT52 terminal. In VT52 mode, all CØ control characters are allowed, although some are ignored. No C1 control characters or ANSI mode control functions are allowed. The user-defined keys are disabled. Table 4-11 defines the VT52 mode escape sequences. Table 3-3 in Chapter 3 defines the VT52 auxiliary keypad codes. The set-up VT1ØØ (ASCII/UK) default character set applies to VT52 mode as well.

Table 4-11 VT52 Escape Sequences

Escape Sequence	Function
ESC A	Cursor Up
ESC B	Cursor Down
ESC C	Cursor Right
ESC D	Cursor Left
ESC F	Enter "Graphics" Mode
ESC G	Exit "Graphics" Mode
ESC H	Cursor To Home
ESC I	Reverse Line Feed
ESC J	Erase To End Of Screen
ESC K	Erase To End Of Line
ESC Y Line Column	Direct Cursor Address
ESC Z	Identify
ESC =	Enter Alternate Keypad Mode
ESC >	Exit Alternate Keypad Mode
ESC <	Enter ANSI Mode
ESC ^	Enter Auto Print Mode
ESC _	Exit Auto Print Mode
ESC W	Enter Printer Controller Mode
ESC X	Exit Printer Controller Mode
ESC]	Print Screen
ESC V	Print Cursor Line

APPENDIX A
VT22Ø/VT1Ø2 DIFFERENCES

This appendix describes the major differences between a VT1Ø2 terminal and the VT22Ø terminal operating in a VT1ØØ mode.

Difference Area	**VT1Ø2**	**VT22Ø**
LEDs	Programmable	Not Programmable
Alternate Character ROM	Socket for OEM supplied character ROM	Down-line loadable character set
Screen Freeze	NO SCROLL key	HOLD SCREEN key
Printer Port Connector	25-pin	9-pin
Screen Refresh	5Ø/6Ø Hz	6Ø Hz only
Set-up	Display in set-up "bits."	Display in three languages. Different display. Has no effect on software.
RIS function	Performs power-up self test	Does not perform power-up self test
Off-Line/Local	Off Line mode disconnects modem	Local mode does not disconnect modem
Communication	Full duplex and half duplex	Full duplex only. Does not affect software
	Transmit speed limitation of 6Ø characters/sec regardless of baud rate	Optional transmit speed limitation of 15Ø characters second
	Selectable passive or active 2Ø mA	Passive 2Ø mA only

Difference Area	VT1Ø2	VT22Ø
Terminal ID	Responds to primary device attributes with: ESC [? 6 c	Responds to primary device attribute with: CSI ? 62;1;2;6;7;8 c
	Secondary device attribute not supported	Responds to secondary device attribuute with: CSI > 1;Pv; Ø c where Pv is firmware version number

APPENDIX B
ADDITIONAL VT220 DOCUMENTATION

This appendix lists and describes additional VT220 documents that can be ordered from DIGITAL.

Title	Number	Description
VT220 Owners Manual	EK-VT220-UG	The VT220 Owners Manual provides the information needed to operate and maintain the VT220.
VT220 Installation Guide	EK-VT220-IN	The VT220 Installation Guide describes the installation procedure for the VT220.
VT220 Programmers Pocket Guide	EK-VT220-HR	The VT220 Programmers Pocket Guide provides a quick-reference summary of VT220 programming information.
VT220 Pocket Service Guide	EK-VT220-PS	The VT220 Pocket Service Guide provides procedures for troubleshooting and repairing the VT220 to the field replaceable unit.
VT220 Illustrated Parts Breakdown (IPB)	EK-VT220-IP	The VT220 IPB provides an illustrated parts breakdown of the VT220.
VT220 Family Field Maintenance Print Set	MP-01732-01	The VT220 Family Field Maintenance Print Set provides a complete set of VT220 electrical and mechanical schematic diagrams.