

Communications Research Centre

PICTURE DESCRIPTION INSTRUCTIONS PDI FOR THE TELIDON VIDEOTEX SYSTEM

by

H.G. Bown, C.D. O'Brien, W. Sawchuk and J.R. Storey

CRC TECHNICAL NOTE NO. 699-E



**Department of
Communications**

**Ministère des
Communications**

5.

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS
CANADA

PICTURE DESCRIPTION INSTRUCTIONS
PDI
FOR THE TELIDON VIDEOTEX SYSTEM

3. by

H.G. Bown, C.D. O'Brien, W. Sawchuk and J.R. Storey

11 *and others*

(Technology and Systems Branch)

6 CRC TECHNICAL NOTE NO. 699-E

November 1979

OTTAWA

CAUTION

This information is furnished with the express understanding that:
Proprietary and patent rights will be protected.

TABLE OF CONTENTS

ABSTRACT	1
1. INTRODUCTION	2
1.1 ALPHAMOSAIC VIDEOTEX	3
1.2 ALPHA-GEOMETRIC VIDEOTEX	4
2. ENVIRONMENT	6
2.1 COMMUNICATIONS MEANS	7
2.2 DATA BASES	8
3. CODING PHILOSOPHY	8
4. PICTURE DESCRIPTION INSTRUCTIONS	17
4.1 PDI	17
4.1.1 Basic Character Sets	22
4.1.2 Diacritical Signs and Special Characters	23
4.2 PICTURE DESCRIPTORS	23
4.2.1 Point	25
4.2.2 Line	27
4.2.3 Arc	29
4.2.4 Rectangle	32
4.2.5 Polygon	34
4.2.6 Bit	36
4.3 PDI CONTROL	41
4.3.0 Clear	45
4.3.1 Domain	46
4.3.2 Drawing Control	48
4.3.3 Tonal Control	49
4.3.4 Line Control	51
4.3.5 Fill Control	52
4.3.6 Marker	56
4.3.7 Wait	58
4.3.8 Text Format	60
4.3.9 Character Definition	65
5. RESPONSE CODES	66
6. CONCLUDING REMARKS	69

PICTURE DESCRIPTION INSTRUCTIONS

PDI

FOR THE TELIDON VIDEOTEX SYSTEM

by

H.G. Bown, C.D. O'Brien, W. Sawchuk and J.R. Storey

ABSTRACT

The Telidon Videotex system is a method by which information can be accessed from central data bases by the general public. By the use of a domestic home television receiver augmented by a micro-computer controlled interface device a user can access pages of graphical and textual information over a public common carrier communications facility such as the telephone network, a cable television line or the data may be even encoded into unused space in a broadcast television signal.

In order to transmit information to a Telidon terminal, at minimum bandwidth, and in a manner independent of the type of communications channel, a coding scheme was devised which encodes a picture into the geometric drawing elements which compose it. These "Picture Description Instructions" are an alpha-geometric coding model and are based on the primitives of POINT, LINE, ARC, RECTANGLE, POLYGON, point by point BIT encoded images and TEXT encoded as ASCII characters. This document provides a detailed specification of this code as well as a description of the principles which make it independent of communications channel and display hardware.

1. INTRODUCTION

This document presents the details of the Picture Description Instructions (PDI) which are used as the control commands for the Videotex system developed in Canada and undergoing field trials by various organizations and communications carriers. This specification was developed at the Communications Research Centre of the Department of Communications, Government of Canada, and includes both the PDI coding specification as well as the response codes to be used between terminals and the host data base computer in interactive Videotex systems. Where possible, the design constraints and other considerations affecting the choice of the PDIs are also given. A brief explanation of the Videotex concept and how it operates over various communications media is presented.

Videotex is a name used internationally to represent a class of home and business information services which disseminate information from public information suppliers into the home. The system essentially makes use of the home television set as a still picture display medium where the consumer has control over what is displayed. An electronics module is added to the home television set to allow it to assemble and display an image made up of characters and graphics drawings. The data to be displayed is received either from a data connection over a telephone line or encoded in the unused flyback scan lines of an over-the-air television or cable television signal.

Interactive Videotex is a service based on the use of dedicated two-way communications lines between a Videotex terminal and a central data bank. Broadcast Videotex (Teletext) is a service based on the use of one-way communications channels such as over-the-air television or cable television signals where the entire repertoire of data pages is continuously transmitted and where the user's terminal waits for and selects the desired page. Except for the considerations of delay and the size of the database available, both services operate in essentially the same way as far as the user is concerned.

In operation, a subscriber to Videotex would be able to select from a number of pages of material which he may wish to display and view. An information supplier would provide a central data bank of information covering a wide range of topics. A subscriber selects which information is to be presented by keying the appropriate number into a calculator-like keypad or typewriter-like keyboard attached to his television set. As an example, a user who wishes to know the latest sports scores selects the sports page which then presents him with a menu consisting of a list of sports to choose from. By keying in the menu selection for hockey, the current hockey scores would be presented.

This document describes both the general structure of the Picture Description Instructions (PDIs) and that particular subset which has been implemented in order to conduct field trials of a Canadian Videotex system. The PDI commands are an extensible set of commands in order that future technological developments can be accommodated. The areas in which extensions are planned are outlined in terms of the philosophy of the PDI structure and are discussed in this document.

This document, as of November 16, 1979, represents an agreed standard of the Telidon coding scheme for field trials in Canada, up to January 1, 1982.

1.1 ALPHAMOSAIC VIDEOTEX

There are a number of such public access information systems under development in various countries, especially in Europe, which are designed to operate over different communication lines and on various types of terminals. Unfortunately, the major problem with these systems is that the limitations of the particular display hardware, especially with respect to resolution, are reflected in the communications protocols they utilize. The European public access data systems are based on an alphamosaic approach. The display screen is divided into a fixed number of character positions, typically 24 rows of 40 character positions per row for a total of 960. Each position is fixed somewhat like the squares of a chessboard. Graphics is accomplished by subdividing each character position into six sub-areas in order to define a pattern. The effective resolution over the entire screen for this approach is 72 by 80 positions. This resolution is quite coarse and pictures drawn on a 72 by 80 grid appear very square-edged and rough. Since television audiences have become accustomed to high quality pictures, poor quality graphics is a severe constraint.

The major problem is not that the resolution of these display terminals is low, but rather that it is fixed and unchanging. The data which is transmitted over the communications channel is in the same mosaic format in which it will later be displayed. The far-reaching effect is that the data communications protocol is tied to the hardware resolution limitations of today's display technology. It will not be possible to utilize the improved resolution of future display terminals since the communications data format will have to remain compatible with the installed inventory of early versions of the terminals.

Another important limitation concerns the data bases in the central computer. The description of the data in these data bases will reflect the communications format. If this also is restricted to the 72 by 80 character grid, it would be very costly to upgrade the data bases if the display resolution were ever to be changed.

The chess-board method of describing a picture results in a very inefficient communications code. Rather than describe a line by the co-ordinates of its endpoints, the coding describes each individual picture element along the line and in the background. This overly repetitive and inefficient description places a larger burden on the communications channel than is necessary and therefore increases the cost of the communications component of the system.

The rapid advances in the state of the art in electronics in the past decade have changed the original design constraints for the alphamosaic approach. The cost of memories have dropped tremendously with the result that high resolution pictures may now be stored and displayed on home television sets at a moderate cost. In addition, the introduction of inexpensive micro-computers has allowed sufficient sophistication to be programmed into a terminal so that a high level, efficient communications code may be devised.

1.2 ALPHA-GEOMETRIC VIDEOTEX

The basic design approach for the Canadian Alpha-geometric Videotex system is based on the premise of independence from particular hardware display apparatus limitations such as the resolution of current display techniques. The concept of forward and backward compatibility is of central importance. Forward compatibility means that the communication codes must be designed in such a way that future terminals will be able to access old data. This allows for growth and means that a future terminal with higher resolution can accept low resolution or otherwise limited data from an established data base. Backward compatibility, which is much more difficult to achieve, means that an installed inventory of terminals can receive and decode all future command formats in an intelligent manner. Again using resolution as an example, a picture which is communicated in high resolution should appear as a low resolution picture on older or less expensive terminals and as a high resolution picture on those terminals equipped to handle high resolution.

The NTSC colour television signal format is a good example of a design for both forward and backward compatibility. At one time only monochrome television sets and signals existed, but now the situation is more complex. Colour sets can receive both black and white signals as well as colour signals (forward compatibility). Monochrome television sets can receive black and white signals, and when receiving colour signals, they interpret them as black and white (backward compatibility). Technically, this compatibility was very difficult to achieve with television signals, but it has allowed the introduction of colour television without the need to replace the installed inventory of television sets. The same type of growth potential must be designed into the Videotex communications protocol from the start.

For the above reasons, the Picture Description Instructions have been defined in terms of basic alphanumeric and geometric primitives such as POINT, LINE and RECTANGLE which describe the structure of the entity to be drawn. For example, a line is drawn by specifying its endpoints. It is the responsibility of the terminal to decode this high level description and to draw the best line possible between the two endpoints. On a high resolution display, finer increments are used to draw the same line that would be displayed in a coarse manner on a low resolution display. In this way, both forward and backward compatibility is achieved because drawing commands are described in terms of geometric primitives rather than in terms of some parameter of the display hardware. The accuracy with which coordinate positions are specified to describe these geometric primitives is reflected in the communications code, but this can be dealt with in a compatible manner by allowing the resolution of this description to be varied and by truncating the coordinate description to the accuracy which can be handled by the terminal.

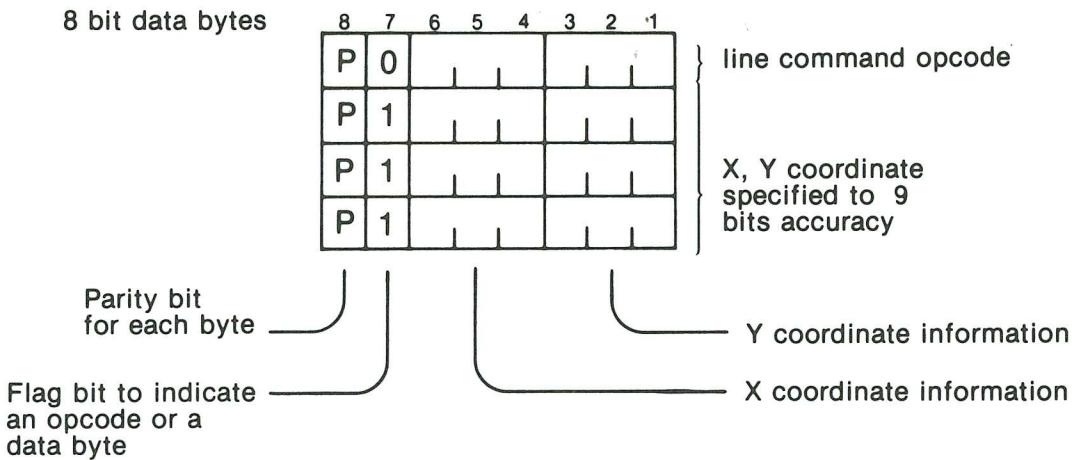
The term "Alpha-geometric" is used in general to describe drawing information encoded in the Picture Description Instruction format. There are two types of drawing descriptions within the PDI codes. These are the descriptions in terms of the geometric drawing primitives of POINT, LINE, POLYGON, etc. properly called "Alpha-geometric" and the "Alpha-photographic" description of an image in a point by point bit encoded manner.

The details of the Picture Description Instructions will be laid out in more detail later in this document. A cursory outline is presented here so that the operation of the coding structure can be examined.

The Picture Description Instructions (PDIs) are a compact set of commands for describing pictures. Each command consists of an opcode followed by a number of bytes of data. These commands are:

- | | |
|-----------|---|
| TEXT | - draw a string of characters |
| POINT | - set the drawing position and optionally draw a point |
| LINE | - draw a line based on its endpoints |
| ARC | - draw a circular arc based on a three point definition |
| RECTANGLE | - draw a rectangular area of specified width and height |
| POLYGON | - draw a polygonal area based on a series of vertex points |
| BIT | - draw an image point by point or encoded in a similar manner to the operation of a facsimile machine |
| CONTROL | - provide control over the modes of the drawing commands. |

As an illustration, the command to draw a line from the current drawing position to a specified location is presented below. The command format is described on a data byte by byte basis to show the amount of information which is specified to describe the x and y coordinates.



The coordinate system for the description of geometric drawing primitives is based on a Cartesian number system ranging from 0 to 1 over the visible area of the display screen. This is independent of the physical resolution of the apparatus which may be a television set with on the order of 256 positions of resolution in the horizontal direction, or a high resolution display apparatus of 1024 positions or any other resolution. Since coordinate positions are specified as a fraction of the width of the display screen, the least significant bits may be dropped when they are not needed.

For the purpose of transmission, the coordinate information has been organized with 3 bits of x and 3 bits of y data per byte. The default specification is to 3 bytes. Additional bytes may be used for coordinate specifications to higher resolution.

The other two areas which must be addressed in order to insure terminal independence in the communications coding format are the effects of the communications channel and various different input devices. The problem of accommodating a variety of input devices in a compatible manner can be handled by placing the various types of interaction devices into a number of classes.

Theoretical work in interactive techniques with display devices has indicated that there are a small group of interactive functions which are performed regardless of the hardware realization of the input device. Basically these functions are:

- | | | |
|-------------|---|---|
| KEYBOARD | - | Input of character string via a keyboard or similar device. |
| PUSH BUTTON | - | Input of specific commands via special function push buttons. A keypad is an example of a push button device. |
| LOCATOR | - | Input of an X, Y position such as the coordinate of a cursor set up by a control arm (JOYSTICK) or a position indicated on a tablet device. |
| IDENTIFIER | - | Input of a menu selection such as the selection of a particular item from a table of choices. |

Any type of input device may be incorporated in a terminal by the terminal manufacturer, but if all communication back to the host computer is in terms of one of the above categories, then compatibility and terminal independence will result.

2. ENVIRONMENT

A Videotex system consists of a central data store, a communication system to distribute the data, and a terminal to interpret and display data requested by a subscriber. Because there will be a large number of terminals in many locations such as homes, offices and schools, the terminal is the most important component of the entire Videotex system both with respect to capabilities and costs. It is envisaged that a number of Canadian manufacturers will produce terminals for Videotex of varying levels of sophistication. A high quality terminal capable of fine resolution and multicolour operation may find use in an office situation, while a low cost minimum feature terminal would be of greatest demand on the cost sensitive home consumer market. A range of terminals will provide the public with choice. A manufacturer is free to produce any level of sophistication in a terminal he desires as long as the terminal is fully compatible with the PDI communication code. The various levels of implementation of certain features such as resolution, character sets, filling of rectangular areas and polygons, etc., will be described later.

The communications system transports picture data in the form of PDIs from the central data base to the consumer's terminal. There are many different methods of communication available; to name a few, there are voice grade telephone lines, cable television lines, optical fibre communication channels, over-the-air broadcast television and digital communications lines such as

packet switched channels. These various types of communication channels vary widely with respect to error rate, data rate, bandwidth and the capability to reverse the channel to transmit control commands back from the terminal to the host data base computer. Although the Picture Description Instructions themselves do not change for different communications media, there are differences in the manner of receiving PDIs, in error coding, in data flow control and in transmitting interaction commands back to the host data base computer. These differences must be considered in the specifications for the interface of a terminal to a communication line.

For the purpose of this report, the relative merit of various communications means will not be discussed, although the existence is acknowledged of important systems factors, such as the loading on telephone exchanges and error susceptibility in cable and off-air systems. Rather, scenarios will be presented for a few communications means in order to illustrate the terminal and interface considerations.

2.1 COMMUNICATIONS MEANS

The manner in which Picture Description Instructions are communicated to the home or business terminal has no effect on the PDI commands themselves but does effect the form of the interactive dialogue for the user. Communications over a narrow bandwidth voice grade telephone line is dedicated with one line per terminal so the response to interactions by the user is very good. However, images take time to build up because the maximum data transfer rate is usually 120 characters per second. Communications over a subchannel of a cable television system offers a high bandwidth line into the home, but the line is a broadcast channel into all homes with the result that statistical queueing of requests is required. Response to interactions would be slow but pictures would build up very quickly. Some cable television systems provide a reverse channel for two-way operation, but if this is not available another channel over a different media would be required to transmit interaction commands back to the host data base computer.

Several unused lines in the flyback period of an over-the-air broadcast television signal may be used to transmit images by encoding the drawing commands in this unused bandwidth. This technique is commonly called Broadcast Videotex or TELETEXT, and can be readily implemented by using PDI drawing commands. Because no reverse channel is possible over the air, a continuous stream of a small number of pages is sent in a round robin sequence. A terminal waits until the page it has requested is transmitted and then captures the page and displays it. The limitations are that only a small number of pictures may be accessed and long delays may be incurred in waiting for a picture to be presented.

Other communications means such as digital data services over the telephone local loop and optical fibre data channels are also capable of transmitting PDIs to home or business. Of particular interest are combined services, for example, the transmission of individual pictures over a circuit switched media such as a telephone line and the transmission of large blocks of data, whole "magazines", over a broadcast facility such as a cablevision sub-channel for later off-line use.

The affect on the terminal hardware of these various communications means is minimized by the use of a common set of Picture Description Instructions. The task of one portion of the terminal is to interpret and execute PDI commands regardless of the communications means. A module to handle each of the possible communications means can interface to the rest of the terminal via a standard RS-232 interface.

2.2 DATA BASES

Vast data bases of pages to be presented on a Videotex terminal would be stored on central computers run by information supplier companies. The cost of establishing and maintaining these data bases will be a major portion of the cost of an entire Videotex system. If PDI codes are used as the communication protocol, then compatibility with various types, generations and resolutions of terminals is achieved. Terminal compatibility is important but there are other valuable uses for PDI codes.

The Picture Description Instructions presented in this document are a compact form of describing pictures and, therefore, are an ideal method of encoding pictures for the data base. Storing PDIs in the data base also simplifies the task of the host computer because all that needs to be done upon an interaction is to select the proper page of information and to dump the stored PDIs to the output channel.

Pages of information in the data base must be generated and maintained by the information supplier company. The fact that the PDI commands are based on geometric primitives such as LINE, ARC and POLYGON means that generation and editing of pictures is also based on geometric primitives. A LINE, POLYGON or other primitive may be added to, deleted from, or otherwise modify a picture. If two areas overlap and one is deleted or moved, the background description is still present in the PDI file. Powerful graphic editor programs can be easily written so that the information supplier can manipulate his data with a minimum of effort. Furthermore, because the PDIs are described in terms of endpoint and vertex coordinates, mathematical transformations such as translation, rotation and scaling can be provided.

It is envisaged that the manipulation of pages of information for the data base will be a major housekeeping problem for an information supplier. This is compounded, if in fact, the Videotex pictures are stored in such low level mosaic code that they are difficult to manipulate. The use of the PDI commands in the data base provides the Canadian system both with flexibility in manipulating pictures and with compact data storage.

3. CODING PHILOSOPHY

The Picture Description Instructions comprise the communications coding structure for the Videotex terminal. This coding system and the Videotex terminal have essentially three modes of operation, which are:

- alphanumeric — characters and numbers
- geometric — geometric primitives of POINT, LINE, ARC, RECTANGLE and POLYGON
- photographic — facsimile-like operation describing an image in a point by point encoded manner.

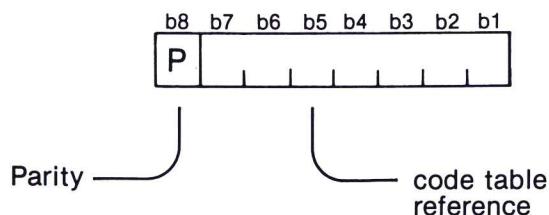
The PDI coding comprises both the "alpha-geometric" and "alpha-photographic approaches.

The alphanumeric mode is the basic mode of the system and it may be used to present alphanumeric data or to annotate graphical drawings. This mode is considered the default mode so that a Videotex terminal in its initial state will be capable of operating as a conventional alphanumeric business terminal.

The geometric mode allows drawing to be presented in terms of the geometric drawing primitives of POINT, LINE, ARC, RECTANGLE and POLYGON. As was discussed previously, the coordinates of the endpoints and vertices of the geometric primitives are communicated in order to achieve independence from the specific display resolution of a terminal.

The photographic mode of operation allows an image to be described either in a point by point manner or in a runlength or otherwise encoded manner in a similar way to the operation of a facsimile machine. Although it is possible to transmit an entire picture on a bit by bit basis much like a digital facsimile, it is envisaged that this mode would have its greatest use for describing small sub-areas of the display screen. For example, in electronic banking an invoice form may be displayed and a sample signature drawn on the screen. Even real photographic images can be handled on a point by point basis.

The international standard ISO 646 on character sets and ISO 2022 on code extension techniques have impact upon the Picture Description Instructions. These standards describe the internationally accepted manner by which alphanumeric data is encoded within an 8 bit wide data field. Bits are numbered b1 to b8 with b8 occupying the most significant position. The bit b8 is either left blank or used to describe parity, while the other seven bits are used as an index to a character code table.



The character code table is normally represented as a table of eight columns and sixteen rows with bits b7, b6 and b5 addressing the columns and b4, b3, b2 and b1 addressing the rows. This table, as illustrated in Figure 2, is subdivided into two areas. The first area occupying columns 0 and 1 contains control characters such as "carriage return" (CR) and "line feed" (LF) and is known as the "C0" set. The second area occupying columns 2

to 7, contains the printable alphanumeric characters A-Z etc., and is known as the "G" set. The basic G set is GØ which contains the specific set of printing characters set out in ISO 646.

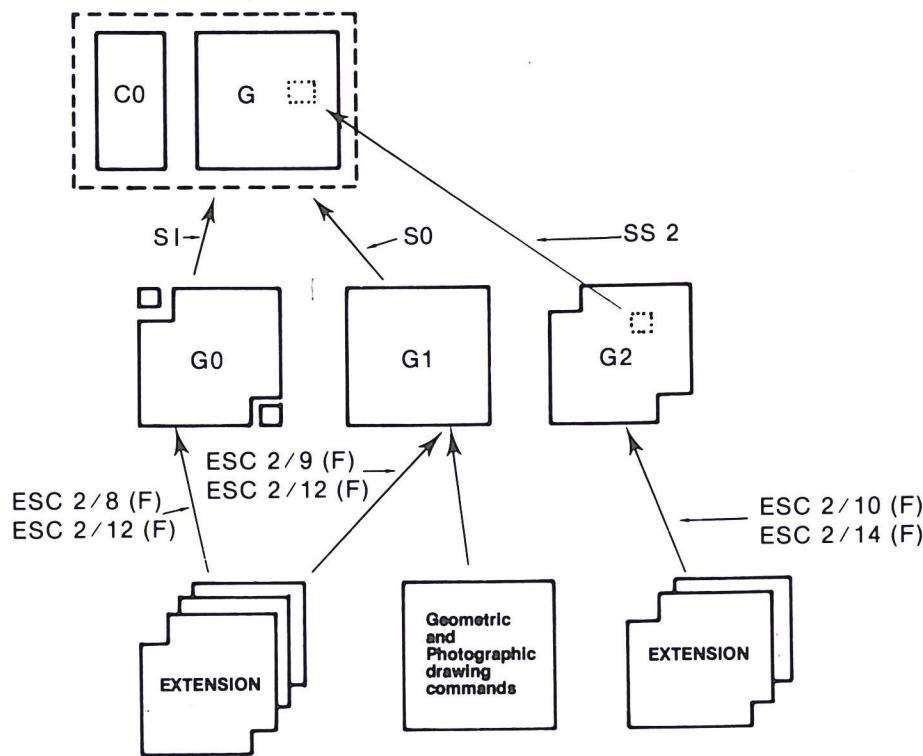


Figure 1. Principles of Code Extension for the Canadian Videotex System

The standards on the 7-bit code and on code extension techniques are contained in ISO 646 and ISO 2022. The Picture Description Instructions make use of the standard code extension technique for invoking alternative meanings for code table positions. The standard CØ control set is always used by the Picture Description Instructions but the G set is switched as the PDI commands change from alphanumeric character to picture descriptor (geometric/photographic mode).

Upon initialization of a Videotex terminal the GØ set of printing characters is established as the current G set of code table interpretations. The control character S0 (position Ø/14 in the CØ set) is used to invoke the G1 set of code table interpretations of geometric/photographic drawing commands. This set of interpretations remains current until an SI control character (position Ø/15) is used to re-invoke GØ. The character S0 and SI operate to toggle between alphanumeric and geometric operating modes.

A third code table G2, which contains accents, special characters, and diacritical signs, exists in the design but will not be implemented for the field trial due to the fact that international agreement as to the composition of the G2 set has not been achieved. Specific codes from this code table will be invoked by the control character single shift SS (position 1/9 in CØ as

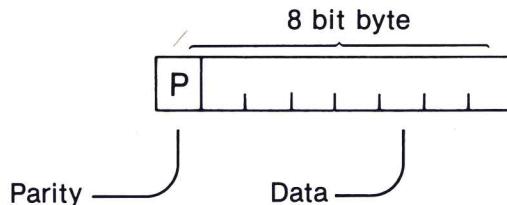
per the proposed revision to ISO 2022 standard). After invocation of this set for a single code interpretation, the current set of G \emptyset or G1 is re-established.

Alternative character sets such as particular national sets or sets for non-Latin alphabet based languages may be designated by an escape sequence. The sequence character ESC (position 1/11) followed by the character in position 2/8 or 2/12 followed by a character in columns 3 to 7 is intended to designate an alternative G \emptyset set. Similarly, the escape sequence (ESC) (2/10) or (2/14) (F) will designate an alternate G2 set. The principles of code extension for the PDI codes are presented in Figure 1. This diagram shows the code table used to interpret codes transmitted to a terminal in the form used for international standard specifications. The C \emptyset portion of the table contains the control characters such as carriage return while the G portion of the table contains the printing or drawing graphics. The second level shown on the diagram presents alternate code table interpretations for alphanumeric (G \emptyset), drawing (G1), and accents and diacritical signs (G2) which are entered into the G portion of the table by the issuance of the characters SI, S0, or SS2 respectively. The third level of the diagram contains descriptions of various character and drawing command sets which may be entered into G \emptyset , G1 or G2 by the escape sequences illustrated. Since escape sequences to all of the required character sets are not yet agreed upon internationally, none of these escape sequences will be implemented for the field trial; rather, the PDI control commands will be used to select character sets.

The International Reference Version of G \emptyset according to ISO 646 is presented in Figure 2.

The G1 set of interpretations for the code table consists of sequences of codes to describe graphical operations. The G1 set is subdivided into two fields, one for operation codes (opcodes) and the other for the numeric data associated with an opcode or for a field of sub-commands accessed via one of the CONTROL opcodes. (See Figure 3.) A code sequence to perform a drawing operation consists of an opcode to draw a POINT, LINE, ARC, RECTANGLE or POLYGON, followed by a variable length sequence of data bytes used to encode numeric parameters for the command. This numeric coding is formed out of codes from table columns 4, 5, 6 or 7, and consists of the least significant six bits of each byte for these code table positions. The Status Commands are dealt with in Section 4.3.

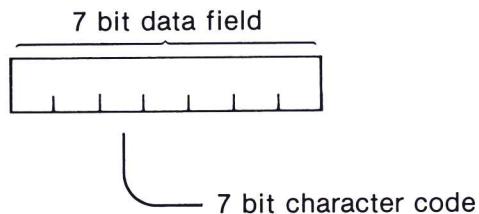
Another way of interpreting the G1 code table assignments is to examine the bit patterns of the codes. A PDI code consists of an 8 bit data byte including a parity bit for error checking. There are, therefore, seven bits of data and one of parity in each byte.



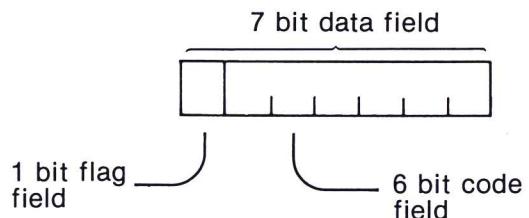
b_7	0	0	0	0	1	1	1	1
b_6	0	0	1	1	0	0	1	1
b_5	0	1	0	1	0	1	0	1
	column				0	1	2	3
		row			4	5	6	7
b_4	b_3	b_2	b_1	0	NUL	TC ₇ (DLE)	SP	0
0	0	0	0	1	TC ₁ (SOH)	DC ₁	!	1
0	0	0	1	2	TC ₂ (STX)	DC ₂	"	2
0	0	1	0	3	TC ₃ (ETX)	DC ₃	#	3
0	1	0	0	4	TC ₄ (EOT)	DC ₄	¤	4
0	1	0	1	5	TC ₅ (ENQ)	TC ₈ (NAK)	%	5
0	1	1	0	6	TC ₆ (ACK)	TC ₉ (SYN)	&	6
0	1	1	1	7	BEL	TC ₁₀ (ETB)	'	7
1	0	0	0	8	FE ₀ (BS)	CAN	(8
1	0	0	1	9	FE ₁ (HT)	EM)	9
1	0	1	0	10	FE ₂ (LF)	SUB	*	:
1	0	1	1	11	FE ₃ (VT)	ESC	+	;
1	1	0	0	12	FE ₄ (FF)	IS (FS)	,	<
1	1	0	1	13	FE ₅ (CR)	IS (GS)	-	=
1	1	1	0	14	SO	IS (RS)	.	>
1	1	1	1	15	SI	IS (US)	/	?

Figure 2. ISO 646 Character Code Table

There are two formats for encoding commands into the remaining seven bits in a byte, one for character codes taken from the G₀ or G₂ set and the other for drawing commands and associated numeric data taken from the G₁ set of code table interpretations. The format for character data is a seven bit character code with which to access the G₀ or G₂ code table.

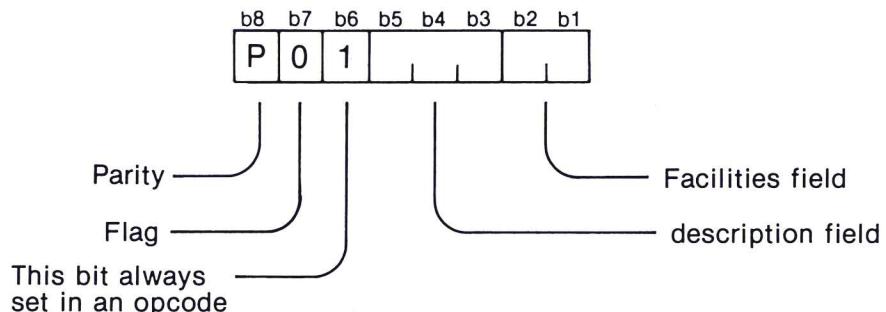


The format for PDI drawing commands is a six bit code field and a one bit flag field.



The flag field of the command is used to indicate whether the byte represents a command opcode or data associated with the command. The flag field is 0 for opcodes and 1 for numeric data. The number of bytes in the code sequence associated with a particular drawing command is determined from the flag field. A command's domain begins by its opcode byte and terminates either by the start of the following opcode byte or by an SO, SI, SS or ESC character.

The field of the opcode is further subdivided into a descriptor field and a facilities field. The descriptor field contains the numeric identifier of each of the eight possible opcodes. The remaining two bits in the facilities field are used for describing optional forms of the instruction. The flag bit (bit 7) equals 0 to indicate an opcode and bit 6 equals 1 to restrict opcodes to columns 2 and 3 of the code table.



There are a possible total of eight opcodes, each having four variants defined by the two facility bits. The eight opcodes are: POINT, LINE, ARC, RECTANGLE, POLYGON, BIT, CONTROL and one spare. The first five of these are geometric picture descriptors and are associated with the alpha-geometric mode of PDI. The BIT opcode is associated with the alpha-photographic mode. The functions of the opcodes are briefly summarized below:

- | | |
|-----------|---|
| POINT | — sets the drawing beam to any position in the display space and optionally draws a point. |
| LINE | — draws a line based on its end points. |
| ARC | — draws a circular arc based on the endpoints of the arc and a point on the arc. The endpoints of the arc may optionally be joined by a chord and the area so defined filled in. |
| RECTANGLE | — draws a rectangular outline or fills in an area of specified length and width. |
| POLYGON | — draws a polygonal outline or fills in the circumscribed area based on a series of defined vertices. |
| BIT | — draws an image point by point or otherwise encoded, in a similar manner to the operation of facsimile equipment. |
| CONTROL | — provides control over the modes of the drawing commands. One of its major functions is to set up a value or colour of an object. Another major function is to access a field of Status Commands via the CONTROL (STATUS) opcode. The Status Commands are essentially sub-commands of the opcode. There is a possible total of 64 Status Commands. |

There is a SET attribute associated with some of the opcodes. The SET position describes the start of the drawing operation in Cartesian coordinates (x,y), on the screen.

Data to be used with a drawing command immediately follows the opcode byte and is distinguished by having the flag bit set to "1". Any number of pairs of coordinates or other data may follow a drawing command until another opcode command word is recognized. The drawing command will be re-executed for each group of data so that, for example, a series of concentrated lines may be specified by one opcode command and followed by the appropriate number of dx,dy coordinate pairs. Repeated SET and POINT data can be used to point plot a graph and repeated RECTANGLE data can be used to draw histograms. If a SET bit is specified for a drawing command, it is interpreted as being repeated as well and coordinate data must be supplied for the SET for every repetition. Those drawing commands such as POLYGON and BIT which use a variable amount of data, end their data list upon the next opcode.

As stated previously, the data associated with a command is often coordinate information. A Cartesian coordinate or displacement in two complement notation is specified as a signed number, usually specified to 9 bits of accuracy including one sign bit. The origin (\emptyset, \emptyset) of the Cartesian coordinate system is the lower left hand corner of the visible display screen.

PDIS have been defined to be independent of the physical resolution of the display media. This accommodates future technological advancements in display apparatus without obsoleting current terminals and data banks defined at lower resolution. The standard television aspect ratio of 4:3 is assumed for television based services using PDIS.

b_7	0	0	0	0	1	1	1	1
b_6	0	0	1	1	0	0	1	1
b_5	0	1	0	1	0	1	0	1
column	0	1	2	3	4	5	6	7
b_4	b_3	b_2	b_1					
0	0	0	0	0				
0	0	0	1	1				
0	0	1	0	2				
0	0	1	1	3				
0	1	0	0	4				
0	1	0	1	5				
0	1	1	0	6				
0	1	1	1	7				
1	0	0	0	8				
1	0	0	1	9				
1	0	1	0	10				
1	0	1	1	11				
1	1	0	0	12				
1	1	0	1	13				
1	1	1	0	14				
1	1	1	1	15				

CO
CONTROL
CODES

NUMERIC DATA
(FOR OPCODES)
OR
STATUS
COMMANDS

SPARE

RECT

POINT

POLY

LINE

BIT

ARC

CONTROL

1 5

2 6

3 7

Figure 3. Tabular Representation of PDI Allocations

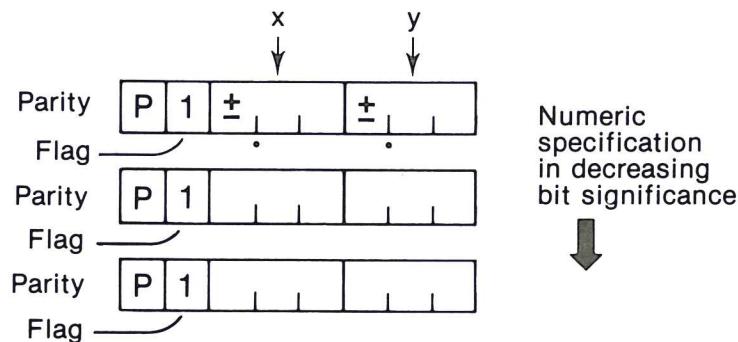
The coordinate specifications are based on a Cartesian \emptyset to 1 numbering scheme. The valid drawing area is entirely visible on the display screen. Any area of the display screen outside of the valid drawing area is termed a "border area" and it is not possible to specify a coordinate position in a border area.

The numbering system is referenced to the visible valid drawing area and consists of coordinates ranging from \emptyset to 1 in both the x and y axes, with positions being specified as fractions of this range. Drawing specifications outside of this absolute area, either directly specified in the negative coordinate space below \emptyset or specified by relative displacements are in error. Any drawing commands or text display operations which cause drawing outside of the valid area are in error.

Coordinate specifications may be described to several levels of accuracy because they are represented as fractions of the visible drawing area. Unnecessary least significant bits are eliminated by truncation when the specification is to a greater accuracy than can be handled by the terminal. The numbering scheme \emptyset to 1 is a single ended open range, that is, the point \emptyset is part of the valid drawing area but the 1 is inaccessible.

Display screens with non-square visible areas map into the square drawing area so that (\emptyset, \emptyset) remains in the lower left hand corner. On a television-like screen with a 4:3 aspect ratio, this corresponds to \emptyset to .99... in the x axis and \emptyset to approximately .75 in the y axis. Drawing commands into the entire square \emptyset to 1 grid are permissible but only the circumscribed 4:3 area is visible. The 4:3 aspect ratio need not be exactly adhered to as it is the entire active television signal which is defined on the 4:3 ratio including the border areas. For example, a manufacturer might build a terminal based on a matrix of 256×200 picture elements.

The default range of the PDI number system is 9 bits occupying 3 bytes of data to describe an x,y coordinate position as illustrated below. It is envisaged that this 3 byte description would be used to communicate the majority of Videotex pictures.



For terminals with greater resolution than can be specified in a number range of 9 bits (>256) an additional data byte may be added to the coordinate specifications extending the range to 12 bits. This number system in no way restricts the terminal manufacturer from implementing any resolution he feels is cost effective at the basic hardware level. The number system

merely describes the manner by which pictures are described so that there is display hardware independence in the PDI code. The manufacturer would convert coordinate information from the PDI coordinate system to his working coordinate system. If his hardware number system bears a simple relation to the PDI number system, this would be a trivial conversion. These coding options are made available so that both information suppliers and terminal manufacturers can make these trade-offs in a compatible manner.

4. PICTURE DESCRIPTION INSTRUCTIONS

Before each of the Picture Description Instructions is presented in a detailed functional manner, it is valuable to address the interrelationship of PDI commands. There is no special positional dependence upon the order in which drawing primitives are presented as there is in the alpha-mosaic coding approach. Pictures coded with PDIs are built up out of a sequence of drawing commands with each geometric primitive superimposed over the previous ones. For example, a RECTANGLE command may be drawn to display an area the full size of the screen as a background. A subsequent POLYGON command would draw over the background RECTANGLE and a subsequent LINE command could overwrite a portion of the POLYGON. In this manner, pictures are built up in layers out of commands from the G1 geometric primitive drawing set.

Manufacturers are free to implement terminal equipment by whatever technologies they deem cost effective. This includes the freedom to implement the G1 graphics drawing display hardware and the G0 and G2 character drawing hardware by different technologies. Because of this freedom the interaction of layering between characters and graphics primitives is deliberately undefined.

In order to provide the terminal manufacturer freedom to build his terminal in the most economic manner, the registration between alphanumeric text, geometric picture, and photographic pictures is not guaranteed to be exact, nor is the superposition of these drawing modes defined. This permits a manufacturer to build an overlaying character hardware generator or to implement characters by software. When a manufacturer varies from absolute registration he is expected to centre any variances about the centre of the visible area.

By not strictly specifying the registration or superposition of characters or bit defined pictures to the geometrically drawn picture, the manufacturer gains certain freedoms in the construction of a terminal. For example, a manufacturer might build a low resolution terminal with 128 pixels of resolution in width. At this resolution it is impossible to define characters within a bit plane memory, so a hardware character generator would be required. This character generator output would be mixed with the geometric picture, the characters in superposition overlaying the geometric picture.

4.1 PDI

The TEXT mode of the Picture Description Instructions is used to encode alphanumeric character strings for display on a Videotex terminal. A seven bit character code is used to describe characters in PDI form. This code is

used as an index to access the GØ and CØ code table interpretations. In its initial state, a Videotex terminal is in TEXT mode in order to maintain compatibility with conventional video and printing business and computer terminals.

There are two formats of operation in the alpha-numeric mode of PDI; one primarily for use in annotating pictorial drawings and the other for use in the general display of textual information. In the first instance, when annotating pictorial drawings, the relationship between characters and pictorial information is very important and must remain exactly as described to ensure that characters used in pictorial drawings remain properly positioned. The second form of the command permits character strings to be displayed in a manner free of geometric constraints. The maximum number of characters and lines of characters per page may be defined for each particular hardware realization of a Videotex terminal. The terminal manufacturer is free to implement whatever character display density he feels is appropriate for a terminal. This allows for the same product diversity in the Videotex terminal marketplace which currently exists with conventional business computer terminals.

In free format mode, format Ø, character size is defined by the terminal, either as a permanent or switch selectable constant. Characters are drawn across the screen to form a line of text. When the margin at the edge of the screen is reached, the drawing position is automatically re-defined to begin at the opposite margin and the vertical drawing position is moved down the screen by a set line spacing. Text may be broken in this manner either on a word or a character boundary. If breaking a line of text on a boundary causes the vertical position to be less than the lower margin, the entire picture is scrolled in a vertical direction by the line spacing.

Most home television sets have significant overscan which means that only a portion of the display raster area is visible. Several studies have been performed, particularly by SMPTE (Society for Motion Picture and Television Engineers) of the overscan on home television receivers. (Ref. P.J. Zavada, Journal of SMPTE, 1974 and C.L. Townsend, Journal of SMPTE, 1957).

According to the SMPTE studies, overscan is significant. Based on the SMPTE studies, it is estimated that 1Ø% margins are required on each side of the television raster area to accommodate overscan. This provides a rather severe constraint on displaying textual information. In order to achieve 4Ø characters in a row, a minimum spacing of one space between characters is required. Since the publication of the SMPTE studies, "ultra rectangular" television screens have become common in home televisions. This somewhat relaxes the overscan constraint, but there have been no recent studies to determine what the overscan limits are on the current population of television equipment. It is felt that television overscan should be investigated, particularly the cases of approximately 1Ø% and 5% borders on each side.

Psychological investigations are being carried out in Canada to determine the readability of various character densities.

The display of 4Ø characters per row is quite acceptable with 5% margins and therefore two picture elements spacing between characters, although with 1Ø% margins, 4Ø characters per row is rather cramped. In the vertical

direction, 20 rows of characters can be displayed with adequate inter row spacing, providing a minimum space for the descender portion of a character. Preliminary psychological investigations have shown that readability is an important parameter. It is also important in Canada to leave sufficient room for accents above lower case characters in French, and study the implications for leaving space for accents in general over any upper case characters.

Due to spacing and readability considerations, it has been proposed that the basic character set for the Canadian Telidon experiments be 40 characters per row and 20 rows per screen. If a terminal is constructed so that an NTSC composite video colour signal at baseband or RF is used to interface a television display monitor, then only 32 characters per row are clearly visible and a character density of 32 characters per row and 16 rows per screen should be used.

Free format text operations would be used in a Videotex system to present long passages of text such as those which occur in a news story. A paragraph would be composed by an information supplier without any explicit "format effecton" control. When the paragraph is displayed on a Videotex terminal the character strings are broken at the margins. On a high character density terminal a number of paragraphs may be displayed, while on an inexpensive low character density terminal only a portion of the text could be displayed at one time. In free format mode the manufacturer is free to implement proportional character spacing or right justification on word boundaries but is not required to do so.

The "format" control characters from the C0 set affect the positioning of text for both annotation and free form text. These characters have the following meanings in free format mode:

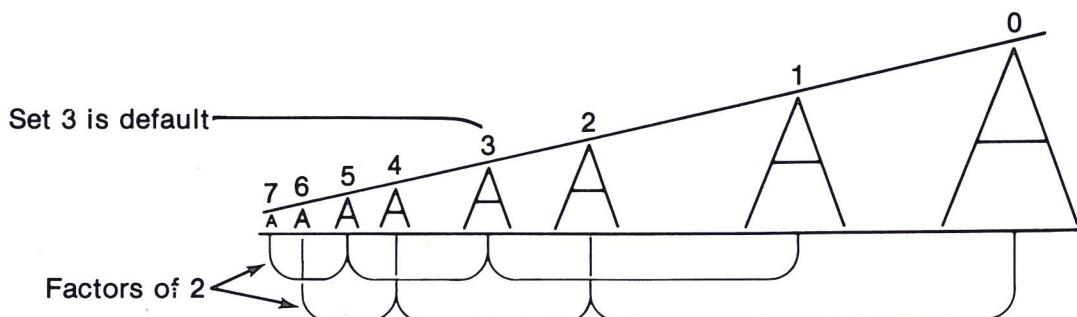
CR — carriage return	— reset the drawing position to the left hand margin X coordinate and the current Y coordinate for free form text (format 0) and reset the drawing position to the X coordinate defined at the last "SET X,Y" operation for annotation text (format 1).
LF — line feed	— reset the drawing position to the current X drawing position and down one line spacing. Scroll vertically by one character height if the vertical position is less than the lower margin.
HT — horizontal tab	— reset the drawing position, advancing it in X to the next tab position. The tab positions are at default set to 1 character width spacing.
VT — vertical tab	— reset the drawing position raising the Y coordinate to the next tab position. Vertical tab positions are default 1 character height (line) but may be redefined by a status command.
BS — back space	— reset the drawing position reducing the X position by one character width. If the left margin is encountered, the backspace operation is ignored.

FF — form feed	— advance to a new drawing form by clearing the display screen and then resetting the drawing position in X to the left margin and in Y to one character height below the top margin. This position is termed the "home" location for the drawing position and a FF performs an implicit SET X,Y command to it.
BEL — bell	— ring the bell associated with the terminal. For those terminals which do not have an audio alarm, this command is ignored.
NUL — null	— no action is performed upon this control character.

The control characters SO, SI and SS are used to change the modes of a Videotex terminal as described previously.

SO — shift out	— exit text mode and enter picture drawing mode, that is, establish G1 as the code table. All data bytes following an SO character are ignored until a command opcode with flag bit set to zero is encountered.
SI — shift in	— enter text mode, that is, establish G0 as the code table. The coordinate position at which characters will begin to be drawn is the current picture drawing position, as executed by the PDI previously while in the SO status.
SS(2) — single shift	— establish G2 as the code table for interpretation of the single character code to follow.

For annotation, format mode 1, characters are displayed according to the spatial and size relationships defined in the character size status command. Characters associated with graphical drawings will, therefore, remain the same relative size regardless of the terminal on which they are displayed. The repertoire of character sizes available is fixed. With due consideration to the cost of implementation of Videotex terminals, a multiplicity of independent character sizes may not be implemented, but a character size, its double, and its quadruple, does not provide a rich enough choice of character sizes for general drawing. Therefore, two base character sizes which vary by a factor of approximately 1.5 are defined and 3 multiples of these characters in factors of 2 have been adopted.



The character sizes are defined in terms of the number of characters which can be displayed on the screen with respect to the number of lines and the number of characters per line. Within the margins specified previously 40 characters per line and 20 lines per page can be achieved on conventional home television sets. The following table describes the range of character sizes which could be achieved.

The following table describes the character sizes:

Size number	#char per line/#lines within margins
0	16/8
1	20/10
2	32/16
3*	40/20*
4	64/32
5	80/40
6	128/64
7	160/80

The default character size is Number 3 marked by an asterisk above. Terminals are expected to support this character size and all larger character sizes (i.e., character sizes 0, 1, 2 and 3).

Character sizes which are too small to be displayed on a terminal of a given physical resolution will be treated as dots. That is, a character of size 7 will appear as:

A on a 1024x800 terminal

■ on a 256x200 terminal

The automatic scrolling and wraparound features supported in character format 0 are not used in format 1. The C0 set control characters LF (line feed), HT (horizontal tab), VT (vertical tab), FF (form feed), BS (back space), BEL (bell), NUL (null) and SO (shift out of text mode), have the same meaning in format 1 as in format 0, except that the automatic scrolling and wraparound features are not used. The carriage return character CR also differs from format 0 in that it resets the drawing position to the current Y coordinate and to the X coordinate defined at the last "SET X,Y" command either imbedded in a previous drawing command or as a separate SET command.

Additional character sets are required for the display of many of the languages of the world. The major scripts are Latin, Russian (Cyrillic), Arabic and the Chinese group. Of particular national interest is the Eskimo phonetic alphabet. Potentially, any number of character sets may be supported by the terminal. The various national alphabets would be given pre-established character set numbers. These character set numbers would have to be

established by international agreement. For the purposes of the Telidon field trials, the two Canadian national standard alphabets (CSA Z234.41) have been adopted as sets numbers one and two as illustrated below, and set number three has provisionally been assigned to the Eskimo phonetic alphabet.

- 1 — Latin alphabet (ISO 646 – CSA Z243.41 set 1)
- 2 — Latin alphabet with French accented letters (CSA Z243.4 set 2)
- 3 — Eskimo and native North American Indian phonetic alphabet

Only set number one, the basic Latin alphabet (ISO 646) is standard and default in all terminals. All others are considered optional, although for field trial purposes both character sets 1 and 2 will be implemented. If a character set does not exist in a terminal, then a reference to that character set number is treated as a reference to character set \emptyset .

A user-defined character set may be made available and is numbered character set \emptyset . Characters may be explicitly defined to suit a user's purposes and transmitted to the terminal. These defined characters or symbols may then be invoked as characters in set \emptyset . This provides a capability for any terminal to display any character or symbol set required. The procedure would be for a user to access a data page which would send the definitions to the terminal, and then to access a data page which requires the alternate character set. If the display is, for example, in the Russian alphabet, references to that character set would be interpreted as references to character set \emptyset , for a terminal which did not support that character set internally. In the event that character set \emptyset has not been set up by accessing the data to be loaded into the terminal, then access to character set \emptyset is interpreted as an access to character set 1. This user definable character set capability will not be implemented for the field trials.

4.1.1 Basic Character Sets

The basic character set which is used in the terminal for the Canadian Videotex system is CSA (Canadian Standards Association) standard Z243.4 - 1973 (set 1) seven bit character set. This set is in general compliance with ANSI standard X3.4 - 1968 and ISO standard ISO 646 - 1963. This set is the default set for the terminal (set Number 1) in order to maintain compatibility with common data processing apparatus. The second set supported by the terminal is CSA Z243.4 - 1973 a (set 2), where those ten characters defined by ISO as being for national purposes are defined as being the 10 most common accented characters in French as spoken in Canada. Additional character sets may be defined for the terminal and software character sets may be loaded into the terminal in the manner described above.

In order to switch character sets, an escape sequence might be executed in compliance with the code extension principles of ISO standard ISO 2022. The escape sequence consists of the character from the C \emptyset set ESC (position 1/11) followed by the character from column 2 row 8, code (if interpreted as a printing character. This is followed by a code from columns 4 to 7 indicating the character set number, up to 63. An additional 63 character codes are

accessible using the character code in position 2/12 as the second character of the sequence. For the purposes of the field trials, only the PDI CONTROL form of character set control will be made available.

The following two tables illustrate character sets 1 and 2, and the Latin alphabet with French accents of CSA Z234.4-2 (Figure 4).

4.1.2 Diacritical Signs and Special Characters

In order to accommodate additional European languages and the Latin alphabet transcription of native Canadian languages, supplementary accents and special characters may be supported.

Accents or special characters may be formed by overstriking characters. The basic set (numbered set 1) contains the characters `!^~, which may be used in combination with the backspace character in order to allow the generation of some accented characters. Because these accents are not sufficient for the diacritical signs in all Latin based alphabets, a single shift mechanism might be used to provide access to an additional group of accents and diacritical signs, as well as to a number of special characters.

Accents are required with many letters including above capital letters in many languages of the Latin script. Of particular interest in North America are the Latin transcriptions of Native North American Indian languages and Spanish where an accent on at least one capital letter is required. In order to accommodate the accents required in these languages, the accent marks might be provided in an augmented character set which may be accessed by the sequence of the control character SS2, from the proposed extension to the ISO 2022 control set CØ, followed by a character from the G2 set. The single shift sequence causes the G2 set to be made current, allows the character to be "printed" and then returns the G set which was current before the single shift sequence.

For the purposes of the field trials the G2 character set has not been defined. There is currently some debate internationally as to whether a G2 character set should contain composite accented characters or only accents for overprinting. Both techniques are compatible with the PDI coding scheme, and either could be implemented when the G2 character set becomes defined by international agreement.

4.2 PICTURE DESCRIPTORS

Invocation of the G1 set of code table interpretations by an S0 control character puts a Videotex terminal in a mode in which it will interpret Picture Descriptor Instructions. In the PDI codes these instructions are sequences of characters which describe geometric or facsimile-like point by point (photographic) drawing commands. Each sequence begins with a code from the opcode portion of the code table and is followed by codes from the numeric portion of the table. Drawing actions activate when there is sufficient information to draw. For example, a series of concatenated line commands may be coded as an opcode followed by pairs of x,y relative displacements. Each of the concatenated lines are drawn when the associated x,y coordinate information is received by the terminal.

SET 1

column	row	b ₀	b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇
0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	1	
1	0	0	0	0	0	0	1	0	
1	1	0	0	0	0	0	1	1	
2	0	0	0	0	0	0	1	0	
2	1	0	0	0	0	0	1	1	
3	0	0	0	0	0	0	1	0	
3	1	0	0	0	0	0	1	1	
4	0	0	0	0	0	0	1	0	
4	1	0	0	0	0	0	1	1	
5	0	0	0	0	0	0	1	0	
5	1	0	0	0	0	0	1	1	
6	0	0	0	0	0	0	1	0	
6	1	0	0	0	0	0	1	1	
7	0	0	0	0	0	0	1	0	
7	1	0	0	0	0	0	1	1	

SET 2

column	row	b ₀	b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇
0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	1	
1	0	0	0	0	0	0	1	0	
1	1	0	0	0	0	0	1	1	
2	0	0	0	0	0	0	1	0	
2	1	0	0	0	0	0	1	1	
3	0	0	0	0	0	0	1	0	
3	1	0	0	0	0	0	1	1	
4	0	0	0	0	0	0	1	0	
4	1	0	0	0	0	0	1	1	
5	0	0	0	0	0	0	1	0	
5	1	0	0	0	0	0	1	1	
6	0	0	0	0	0	0	1	0	
6	1	0	0	0	0	0	1	1	
7	0	0	0	0	0	0	1	0	
7	1	0	0	0	0	0	1	1	

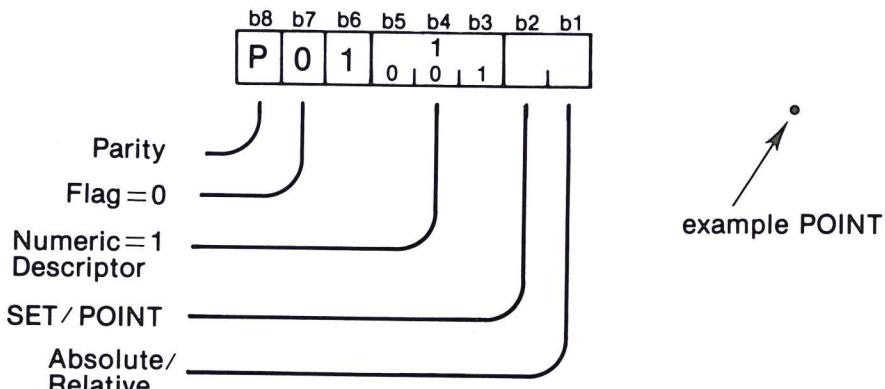
Figure 4. Canadian National Character Sets 1 and 2

There are eight possible opcode coding combinations of which one is used for control of the status of the terminal and another is reserved as a spare. The remaining six opcodes are the geometric drawing commands POINT, LINE, ARC, RECTANGLE and POLYGON, and the point by point (photographic or facsimile-like) BIT command.

There are two facilities bits associated with each opcode to control variations on the commands. In the six drawing opcodes the high order facilities bit (b2) is used to control the absolute positioning SET attribute which may be associated with a drawing opcode. For these drawing commands, the coordinate at which to begin drawing may be explicitly defined or may be taken as the point at which the previous drawing command finished drawing. The exception to this is the POINT opcode in which a new coordinate is always defined, and the drawing position is SET, but the high order facilities bit is used to control whether a point will be drawn or not (i.e. visible or not). The other facilities bit (b1) is used in a separate manner by each command.

In the following subsections each opcode is presented individually.

4.2.1 POINT



The POINT opcode is used to perform the two most basic geometric drawing operations, that of establishing the coordinate at which to commence drawing and drawing a point. An x,y coordinate must always be specified with this command to set the drawing position (SET). Optionally, depending upon the setting of facility bit b2, a point may be drawn (i.e. made visible) at the specified x,y coordinate position (POINT). The x,y coordinate may either be specified as an absolute position or as relative displacement from the previous drawing position.

A series of coordinate positions following a POINT opcode may be used to draw a point by point graph.

OPERATION

- SET the drawing position to the specified x,y coordinate

- If facilities bit b1 equals \emptyset , the x,y coordinate is interpreted to be in absolute coordinates; otherwise, if bit b1 equals 1, the x,y coordinate is interpreted as being a relative displacement to the current position.
- If facilities bit b2 equals 1, a point is drawn (i.e. visible), otherwise if b2 equals \emptyset , no point is drawn and only an x,y SET of the drawing position is effected.
- If additional numeric data follows a POINT opcode the command is repeated.

The four possible forms are:

SET & POINT (absolute)

(Invisible)

POINT absolute
coordinates x, y

SET
co-ordinate

P	0	1		1		0	0
---	---	---	--	---	--	---	---

P	1		x		y		
---	---	--	---	--	---	--	--

P	1						
---	---	--	--	--	--	--	--

•

•

•

SET & POINT (relative)

(Invisible)

x, y

SET
co-ordinate

P	0	1		1		0	1
---	---	---	--	---	--	---	---

P	1		x		y		
---	---	--	---	--	---	--	--

P	1						
---	---	--	--	--	--	--	--

•

•

•

POINT FORMS

SET & POINT (abs.)

(Visible)

x, y

P	0	1		1		1	0
---	---	---	--	---	--	---	---

P	1		x		y		
---	---	--	---	--	---	--	--

P	1						
---	---	--	--	--	--	--	--

SET & POINT (rel.)

(Visible)

x, y

P	0	1		1		1	1
---	---	---	--	---	--	---	---

P	1		x		y		
---	---	--	---	--	---	--	--

P	1						
---	---	--	--	--	--	--	--

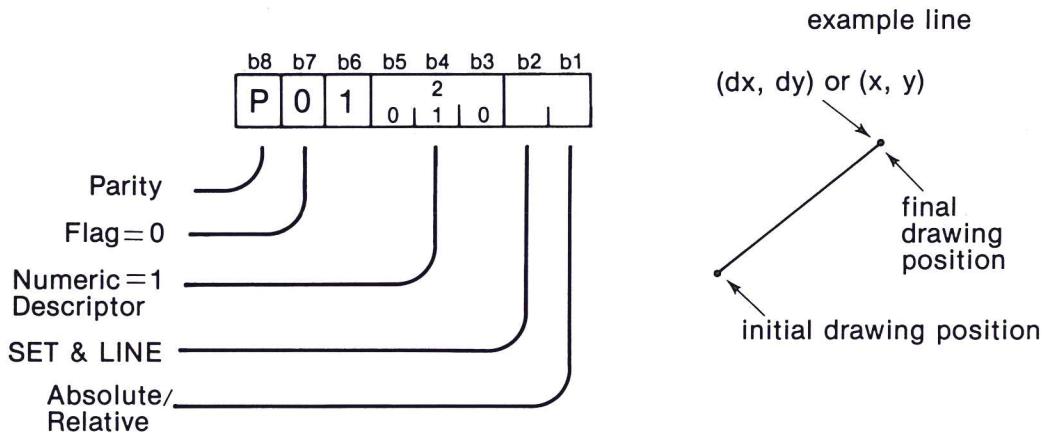
•

•

•

SET
co-ordinate

4.2.2 LINE



The second most basic geometric drawing operation is the LINE opcode. The direction and length of a line is specified by the endpoints. The initial drawing position of a line drawing operation may be either explicitly specified within the LINE opcode or interpreted as being the final position of the previous drawing opcode. The final drawing position for a line segment may be either specified as a relative displacement from the initial position or as an absolute x,y coordinate.

The LINE opcode may be used to draw a line graph from a table of numbers described as absolute or relative coordinates in the same manner as the POINT opcode.

A line may have the attribute texture which is specified by a Status Command accessed via the CONTROL (STATUS) opcode. Line texture may be solid, dotted, dashed or dot-dashed. The default texture is solid.

OPERATION

- If facilities bit b2 equals 1, the x,y coordinate numeric data is interpreted to be an absolute set of the current drawing position; otherwise, if bit b2 equals 0, the current drawing position is interpreted as being the final position of the previous drawing command.
- A line is drawn of the line texture specified.
- If facilities bit b1 equals 0, the x,y coordinate for the endpoint of a line segment is interpreted to be in absolute coordinates; otherwise, if bit b1 equals 1, the x,y coordinate is interpreted as being a relative displacement to the current position.
- If additional numeric data follows a LINE opcode, the command is repeated.

The four forms are:

LINE (absolute)

P	0	1	2	0	0
---	---	---	---	---	---

x1, y1

P	1	x1	y1		
---	---	----	----	--	--

absolute co-ordinate specification for endpoint of line segment

⋮

LINE (relative)

P	0	1	2	0	1
---	---	---	---	---	---

dx 1, dy 1

P	1	dx 1	dy 1		
---	---	------	------	--	--

relative displacement specification for endpoint of line segment

⋮

SET & LINE (absolute)

P	0	1	2	1	0
---	---	---	---	---	---

x, y'

P	1	x	y		
---	---	---	---	--	--

SET co-ordinate

P	1				
---	---	--	--	--	--

P	1				
---	---	--	--	--	--

x1, y1

P	1	x1	y1		
---	---	----	----	--	--

absolute co-ordinate specification for endpoint of line segment

⋮

SET & LINE (relative)

P	0	1	2	1	1
---	---	---	---	---	---

x, y

P	1	x	y		
---	---	---	---	--	--

SET co-ordinate

P	1				
---	---	--	--	--	--

P	1				
---	---	--	--	--	--

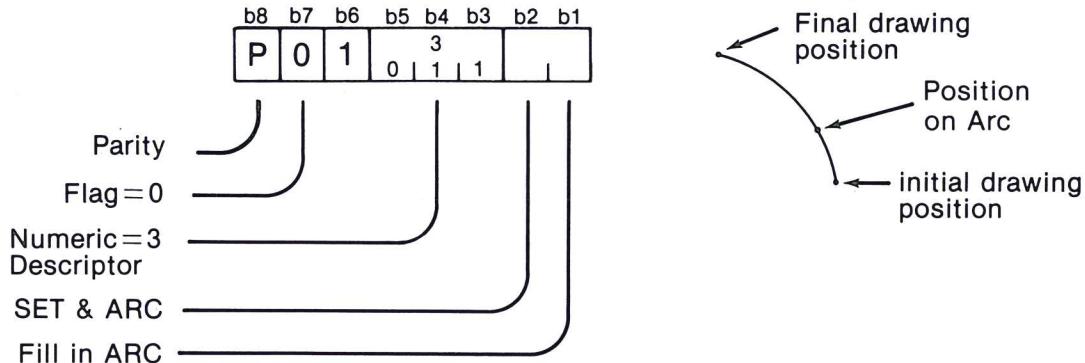
dx 1, dy 1

P	1	dx 1	dy 1		
---	---	------	------	--	--

relative displacement specification for endpoint of line segment

⋮

4.2.3 ARC

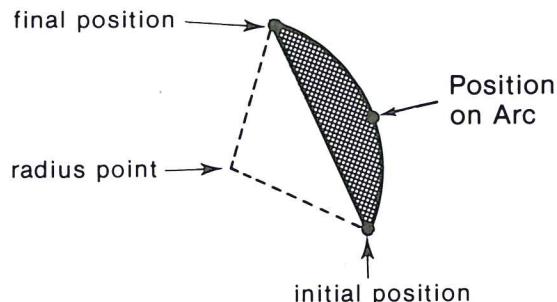


The ARC geometric drawing operation provides the capability of drawing circles and segments of circles. An arc is drawn from an initial drawing position to a final drawing position through a third point on the arc. The initial drawing position may be either explicitly specified within the ARC opcode or interpreted as being the final position of the previous drawing opcode. The position midway on the arc is described as a relative displacement from the initial drawing position and the final drawing position is also specified as a relative displacement from the position midway on the arc. It is good practice, in order to minimize error, to always specify the point on the arc midway between the start and endpoint.

The current drawing position at the completion of drawing an arc is the endpoint specified. Drawing a circle results when the start and endpoint are coincident. For the definition of a circle, the point on the arc defines the diameter of the circle and therefore is the midpoint between the start and endpoint. If the three drawing points are co-linear, a line or lines result from the start to the endpoint through the other point.

An arc may have the attribute of texture which is specified in the same manner as for LINE and other geometric drawing opcodes. Arcs are calculated in as smooth a manner as possible for the display medium.

The area enclosed by an ARC opcode may be filled in to produce circular edged areas dependent upon the setting of a facilities bit. The form of the filled-in area is the area enclosed by the ARC command and the chord joining the two endpoints of the arc.



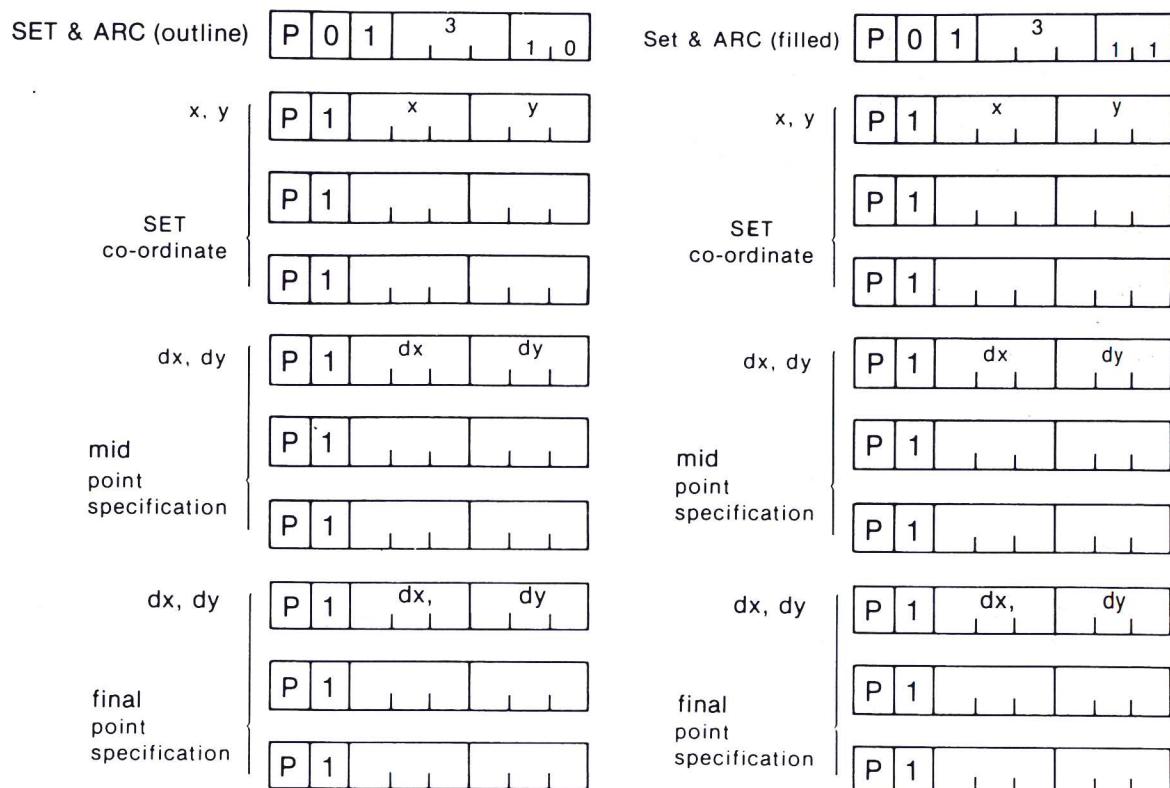
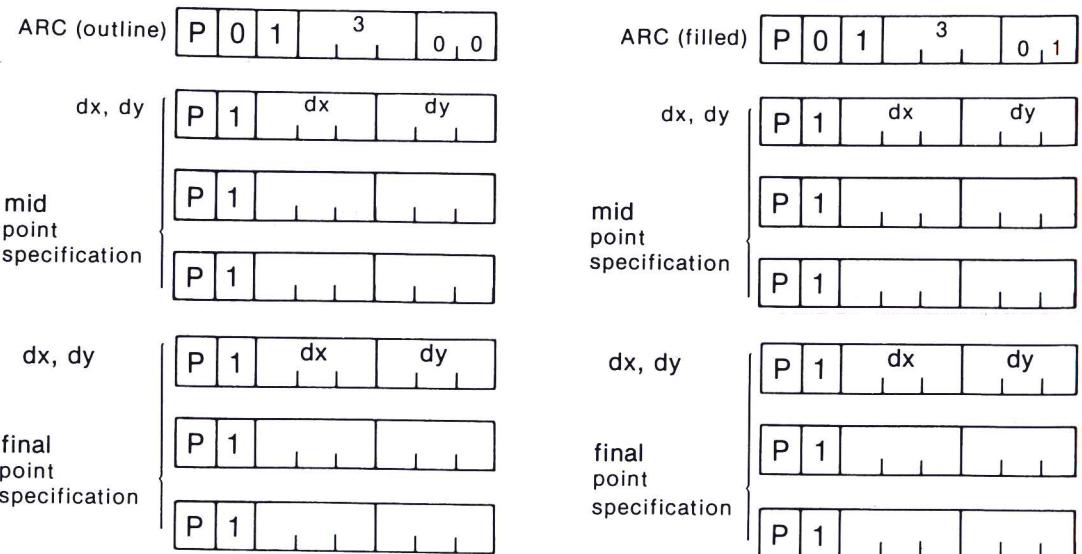
The attribute of crosshatched area filling which is specified by a Status Command applies to filled ARC commands.

The ARC opcode may optionally be used in the future to invoke other arc generating functions such as quadratic spline arcs dependent upon the control status commands. If more than three data points are specified for an arc command, the interpretation is reserved for the future for a higher order arc descriptor function described by a start point, an endpoint, and points on the curve. It may be filled between the curved line and the chord line joining the start and endpoint in a similar manner to a polygon description.

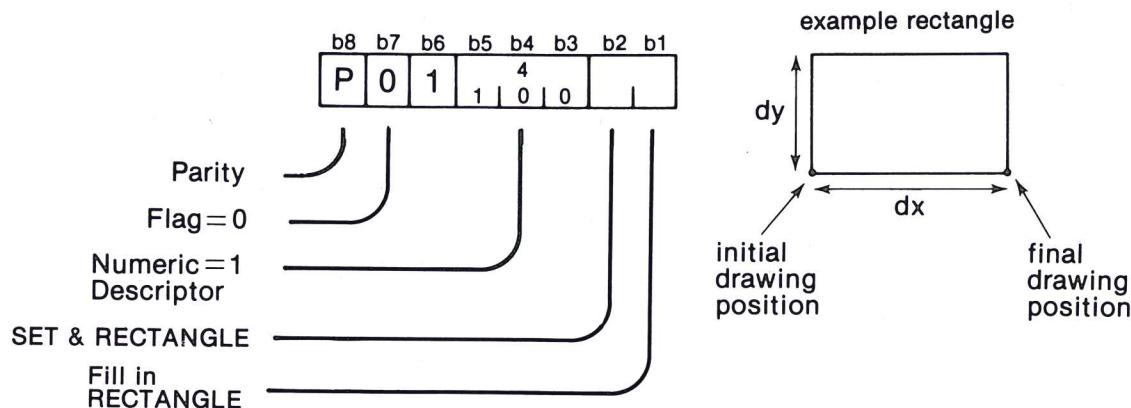
OPERATION

- If facilities bit b2 equals 1, the x,y coordinate numeric data is interpreted to be an absolute set of the current drawing position; otherwise, if bit b2 equals 0, the initial drawing position is interpreted as being the final position of the previous drawing command.
- Coordinates for the position on the arc and the final position are interpreted as being relative displacements.
- If facilities bit b1 equals 0, an arc is drawn from the initial drawing position to the final drawing position through the point specified as on the arc. Line texture applies in a similar manner to the other geometric drawing commands. However, if facilities bit b1 equals 1, the arc defined by the ARC opcode is filled from the chord, with or without crosshatching as specified by a Status Command.
- If a fill operation has been performed, the curved side of the filled area defined by the arc command may be highlighted either in the same colour or in a contrasting colour. This also is controlled by a Status Command.
- If additional numeric data follows an ARC opcode, it is ignored.

The four forms are:



4.2.4 RECTANGLE



The RECTANGLE geometric drawing operation provides the capability of drawing a rectangular area of width dx and height dy. The initial drawing position of a RECTANGLE drawing operation may be either explicitly specified within the RECTANGLE opcode or interpreted as being the final position of the previous drawing opcode. The final drawing position of a RECTANGLE opcode is the initial drawing position altered in x only, by the amount of the dx displacement.

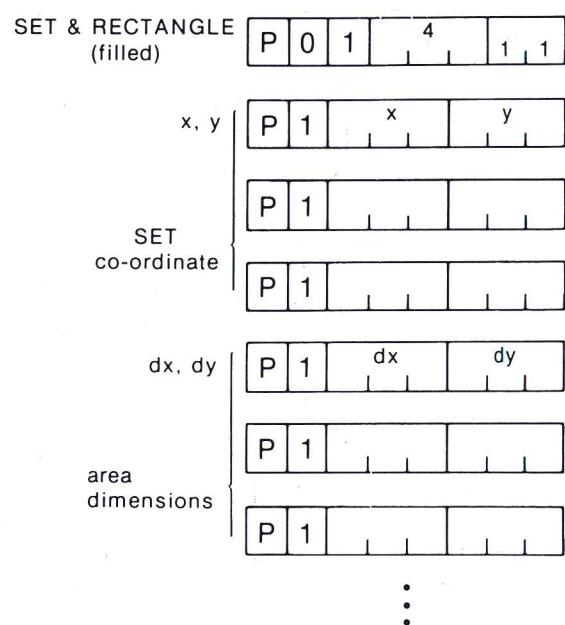
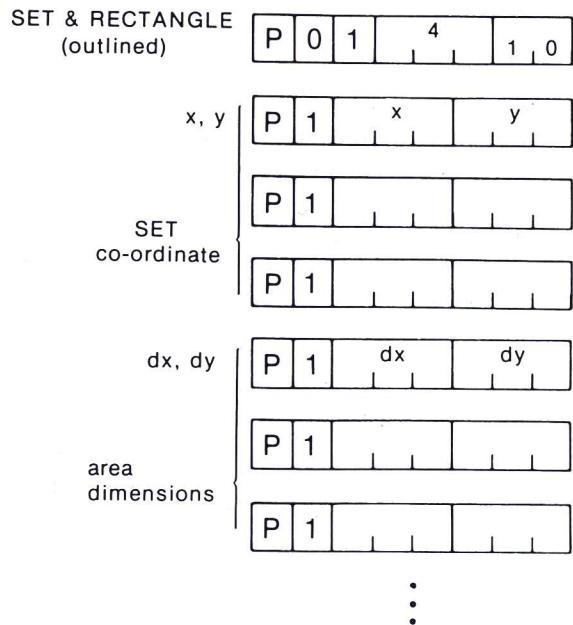
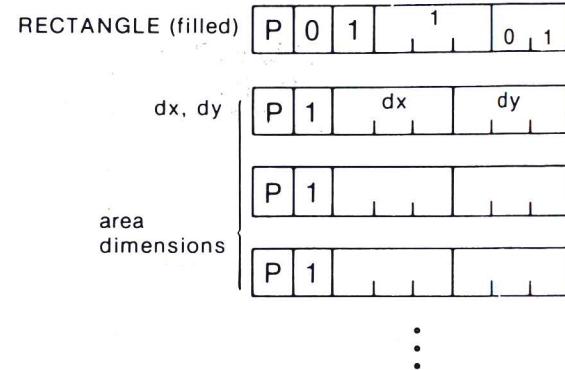
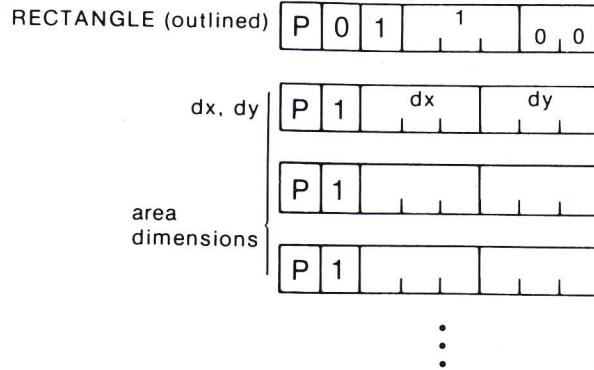
The RECTANGLE opcode may be used to draw a histogram from a table of numbers representing relative dy and dx displacements in the same manner as the POINT and LINE opcode graph plot mode.

A RECTANGLE may be either filled or outlined as specified by a facilities bit. For outlined RECTANGLES, the line texture which is specified by a Status Command applies and may be set to solid (default), dotted, dashed or dot-dashed. For filled in RECTANGLES, the area may be either entirely filled or crosshatched as specified in another Status Command.

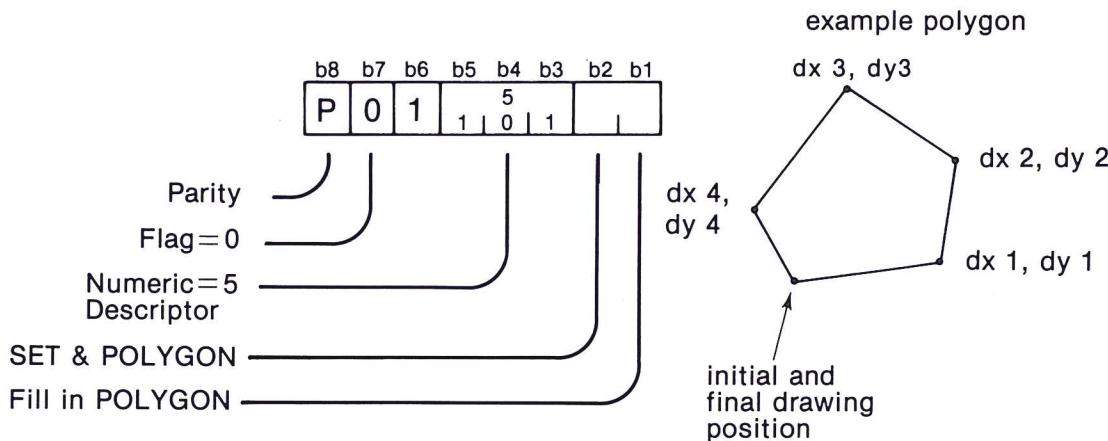
OPERATION

- If facilities bit b2 equals 1, the x,y coordinate numeric data is interpreted to be an absolute SET of the current drawing position; otherwise, if bit b2 equals 0, the initial drawing position is interpreted as being the final position of the previous drawing opcode.
- If facilities bit b1 equals 0, an outline of the RECTANGLE is drawn from the initial point. The final drawing position is always the initial drawing position shifted in x by the dx displacement. Line texture applies in a similar manner as to the other geometric drawing commands. However, if facilities bit b1 equals 1, the rectangular area defined is filled, with or without crosshatching as specified by a Status Command.
- If a filled area has been drawn, the border of the area may be highlighted, either in the same colour or in a contrasting colour in order to outline the filled area. This is also controlled by a Status Command.
- If additional numeric data follows a RECTANGLE opcode, the command is repeated.

The four forms are:



4.2.5 POLYGON



The POLYGON geometric drawing operation provides the capability of drawing a general polygonal area of the specified vertices. A POLYGON is specified as a series of x,y coordinates of the vertices about the perimeter of the polygon. Each dx, dy coordinate pair represents a relative displacement from the last vertex and a relative displacement of magnitude \emptyset is ignored. There is implicit closure between the initial drawing position and the last vertex specified so that the final drawing position is identical with the initial drawing position.

A POLYGON must enclose a single area; that is, no line joining two consecutive vertices may cross any other line joining consecutive vertices. In other words, the polygon specified must not fold over itself.

A polygon may be either filled or outlined as specified by a facilities bit. For outlined polygons, the line texture control status applies. For filled in polygons the area enclosed may be either entirely filled or cross-hatched as specified in a Status Command.

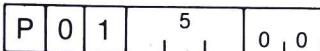
The number of vertices describing a polygon is determined by the amount of data following the POLYGON opcode. The polygon is not drawn until all the vertices have been communicated to the terminal and a termination code such as another opcode or an SI, SO, SS(2) or ESC has been received. The maximum number of vertices permitted to describe a polygon is related to terminal buffer size and is arbitrarily fixed at 256 for the Telidon field trials, and all extraneous data beyond this is ignored.

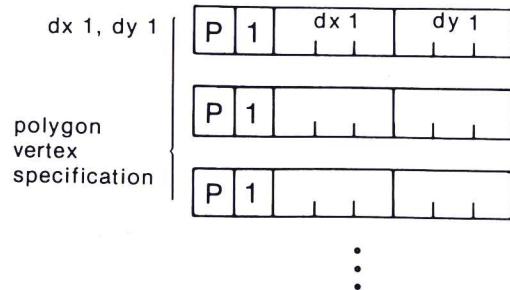
OPERATION

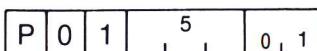
- If facilities bit b2 equals 1, the x,y coordinate is interpreted to be an absolute SET of the current drawing position; otherwise, if bit b2 equals 0, the initial drawing position is interpreted as being the final position of the previous drawing opcode.
- The numeric data representing the vertices of the polygon is accumulated and interpreted as displacements up to the maximum number of vertices or until a termination code such as the next opcode or an SI, SO, SS(2) or ESC is encountered.

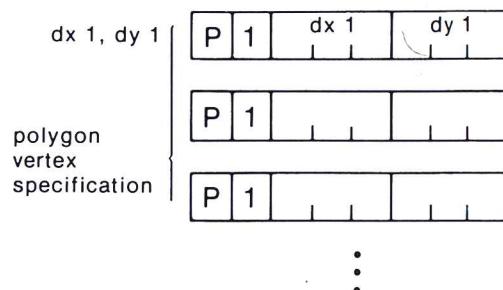
- If facilities bit b1 equals \emptyset , an outline of the perimeter of the polygon is drawn. Line texture applies to the outline as it does for other geometric drawing opcodes. However, if facilities bit b1 equals 1, the polygonal area is filled, with or without crosshatching as specified by a Status Command.
- If a filled polygon has been drawn the border of the polygon may be highlighted, either in the same colour or in a contrasting colour in order to outline the polygon. This is also controlled by a Status Command.

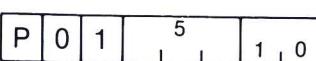
The four forms are:

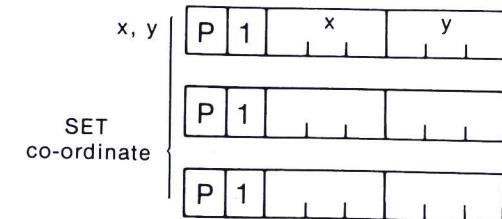
POLYGON (outlined) 

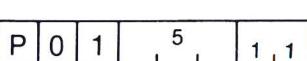


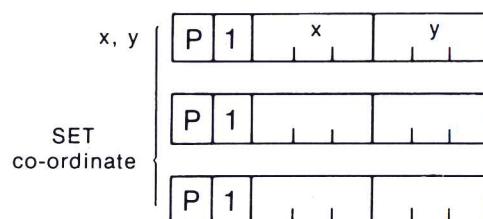
POLYGON (filled) 



SET & POLYGON (outlined) 

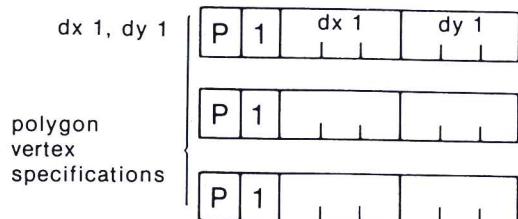
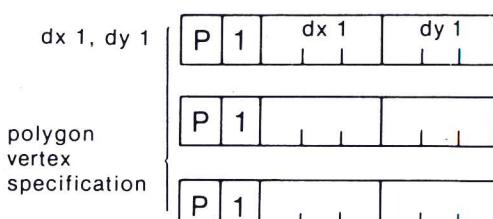


SET & POLYGON (filled) 

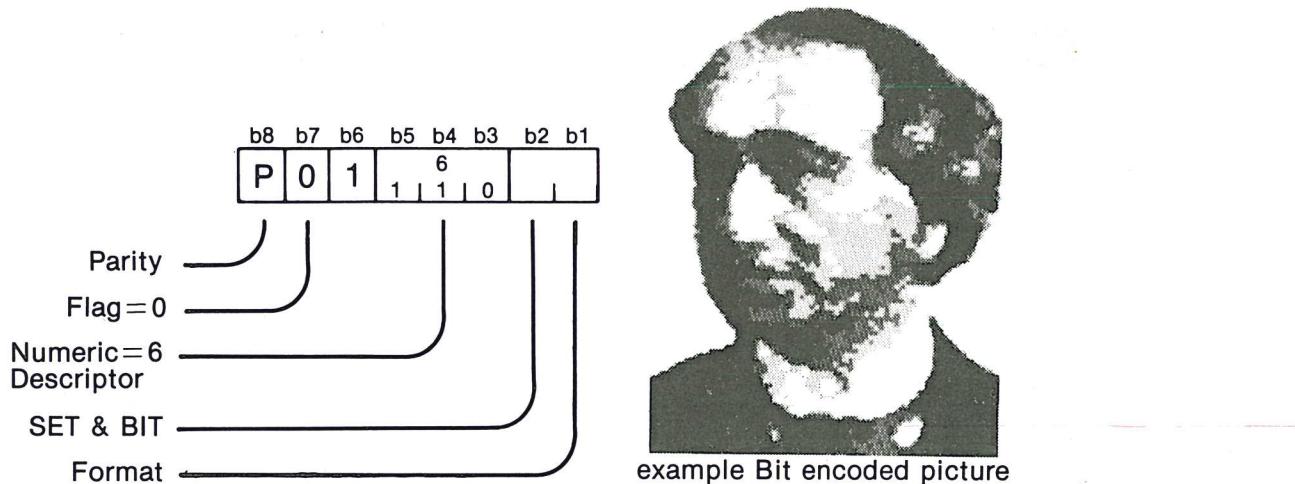


dx_1, dy_1

polyon vertex specifications

4.2.6 BIT



The BIT opcode is a photographic drawing primitive which allows an image to be described in a point by point or otherwise encoded manner in a similar way to the operation of a facsimile machine. A BIT drawing command stores a particular value into a picture memory storage location (Pixel). The value of each pixel represents information about colour or grey scale intensity for a single picture element. A typical hardware display apparatus consists of an array of pixels organized in some manner and of a resolution which varies from low such as 64 by 48 elements in each dimension, to high resolution such as 1024 by 768. The organization of the pixel memory could be a character oriented mosaic approach, or a raster organized bit plane approach or other. The BIT opcode is set up in an independent manner from the particular hardware implementation methods and it is the responsibility of the implementation on each type of terminal to realize the picture described by the BIT opcode as best it can on a particular type of hardware.

The two data formats for the BIT opcode allow for optimum coding for different types of picture information. Those images which consist primarily of line drawings and large areas of constant colour or grey scale are best described by a form of runlength coding, but those images which are photographic like, especially those containing a large number of grey scale changes, are best described on a point by point basis. Both formats allow a series of points to be drawn in display picture memory. The starting point is the current drawing position which may be either explicitly specified in the opcode or taken to be the final position of the previous drawing opcode. The points are inserted in the memory in the order and with the spacing defined, with increasing x coordinate. The maximum x line length specified in the opcode provides a right margin for drawing an area by the BIT command. At the right margin there is an automatic foldover to the left margin described by the current drawing position specified at the beginning of the BIT opcode.

If in describing an image, line by line, down the screen, the lower margin is encountered, the entire displayed picture is scrolled upwards.

If facilities bit b2 equals 1, the starting position is defined within the BIT drawing operation; however, if bit b2 equals 0, no SETpoint is defined and the current drawing position is assumed to be the final drawing position of the last drawing command.

Facilities bit b1 is used to control the drawing format. Format 0 is point by point drawing and occurs when bit b1 equals 0, while Format 1 is encoded drawing and occurs when bit b1 equals 1.

A format 0 point by point drawing instruction has the following format.

P	0	opcode	
P	1	optional x, y SET	
P	1		
P	1		
P	1	x line length	pixel size dxy
P	1	packing	
P	1	value	
P	1	value	
P	1	value	

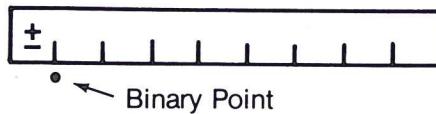
Format 0
Pixel by Pixel
encoded

For a Format 0 point by point Bit drawing instruction the parameters specified have the following interpretations. By specifying the size of the picture area to be drawn in terms of the address field, there is no dependence on the size of the actual picture memory used. After each small square area is drawn for each bit value, the drawing position is incremented in the x direction by the width of the square area (dxy). A value of dxy of 0 or negative is considered illegal. The "X" line length (XLL) is a number describing the maximum dx displacement from the starting position for this drawing opcode. The "X line length" is a positive integer value specifying the number of picture data elements of size dxy which comprise one line of a bit description; that is, if x line length equals 100, then 100 picture data elements will be received to describe a line. The size of the picture data element as it maps into the picture memory elements may not be an exact mapping. This means that any change in size of the basic pixel element due to truncation may result in the registration of BIT pictures and geometric pictures varying in size on some terminals.

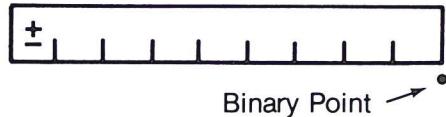
An example follows which illustrates a possible registration error which is acceptable even though it alters the size of a BIT encoded picture with respect to geometric drawing definitions.

Let us say that a bit encoded picture was defined 100 data picture elements wide and dxy was devined as being binary fraction .0000000011 (base 2); that is, on a physical display of 1024 pixels in the x axis each data picture element would occupy three physical pixels. This would mean that on such a piece of display apparatus the picture would occupy .293 (base 10) of the visible screen width 0 to 1. On a display apparatus which has only 512 horizontal pixels, each data picture element would map into one physical picture element. The width of the displayed image would shrink to .195 of the visible screen width and, of course, the picture would drop in resolution by a factor of 2. This is acceptable degradation for this type of a situation as any attempt to compensate for the change in width of the bit picture would introduce serious quantization errors. On a display apparatus which has only 256 physical horizontal pixels, there are too many data pixels for the number of physical pixels so every second data pixels will be dropped, resulting in a picture again of half resolution which is also .195 of the visible screen width.

The integer number specifying (X line length) XLL is specified in the same manner as are binary fractions of 0 to 1 except that the binary point is shifted by the amount defined by the current Domain status. If the current Domain status defines a 3 bytes specification for coordinates, then numbers have the form of sign bit followed by an imaginary binary point followed by 8 bits of data.



in this case the integer number would be

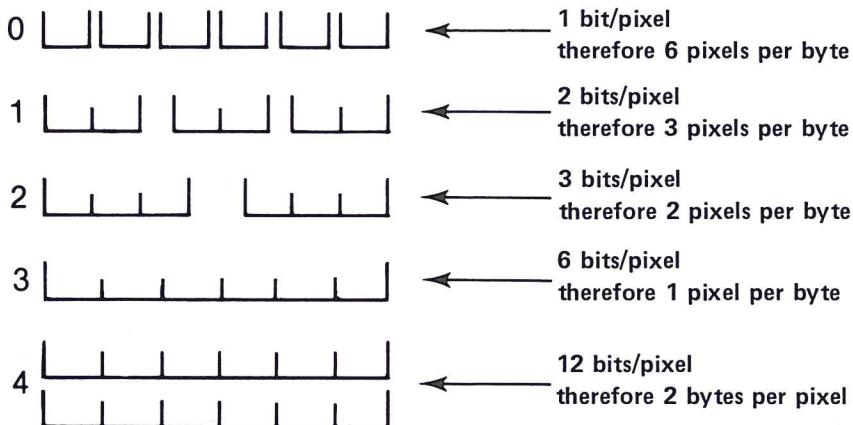


For Domain status specifying more bits of information for the integer specification, the binary point is shifted to the right by an additional number of bits until there is no fractional part to the number; that is, the binary point is considered to be at the left hand end of the number for numbers represented as fractions such as coordinates, while it is assumed to be at the right hand end for integer numbers such as the count of pixels in a line. The default DOMAIN specification is for 3 bytes which is 8 bits plus sign. If "XLL" multiplied by dxy specifies a run of pixels which extend beyond the physical dimensions of the 0 to 1 coordinate space, the extraneous pixels are ignored.

Since the communication of picture information pixel by pixel is costly in communications resources, it is important that information be well packed to minimize the volume of data which needs to be transmitted. Typically, only 3 bits of data need be transmitted for each picture element to represent Red, Green and Blue of the grey scale. If three bits are used for a pixel value,

this nicely packs two pixel values per 6 bit data field in each byte. The "Packing" descriptor specifies the number of pixel bits which will be packed into the six bit wide data bytes of the PDI codes. The number of bits describing a pixel value may be 1, 2, 3, 6, 12 ... or more. Typically, 6 bits per value is sufficient, but additional bytes may be used to provide 12, 18 or more bits per pixel value so as not to artificially limit the format. These formats are presented below:

Packing code



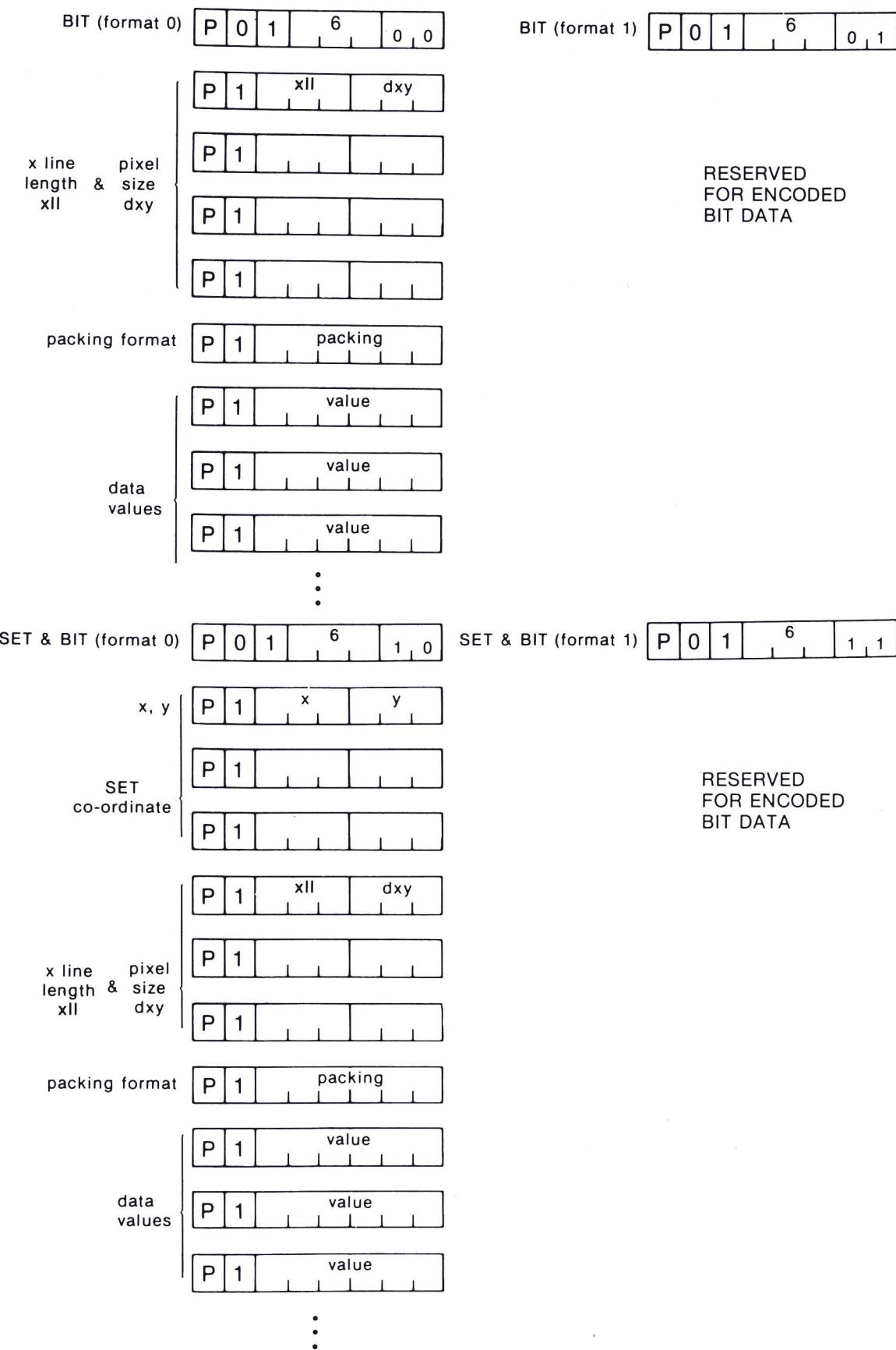
The X line length (XLL) times the data pixel width (dxy) specifies the width of BIT picture. This width may not exactly match a packing density in which case any extraneous data bits specified are ignored.

The format 1 runlength or otherwise encoded drawing instruction which occurs when facility bit b1=1, is reserved for future use and will not be implemented for the field trials. Such a format specification will be ignored.

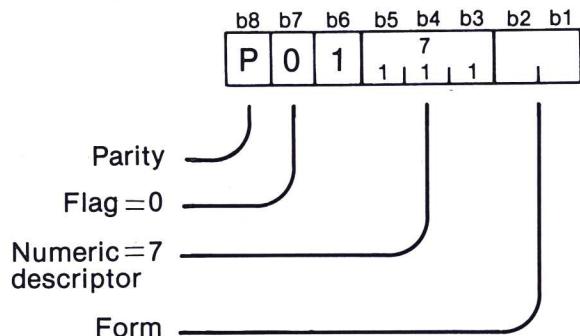
OPERATION

- If facilities bit b2 equals 1, the x,y coordinate is interpreted to be an absolute SET of the current drawing position; otherwise, if bit b2 equals 0, the initial drawing position is interpreted as being the final position of the previous drawing opcode.
- The numeric data next interpreted is in the same form as an x,y coordinate but is actually two numbers which are "x line lengths" and "pixel size dxy".
- Facilities bit b1 indicates which of the two formats for the BIT opcode will be used. This also affects the way in which the next data byte for "packing" will be interpreted. If facilities bit b1 equals 0, the data to follow is in point by point encoded BIT mode, however, if facilities bit b1 equals 1, the data to follow is in runlength or otherwise encoded BIT mode (a reserved mode for future use).
- The remaining data is interpreted in point by point or encoded BIT mode dependent upon facilities bit b1, and terminating upon the next opcode or SO, SI, SS(2) or ESC control character.

The four forms are:



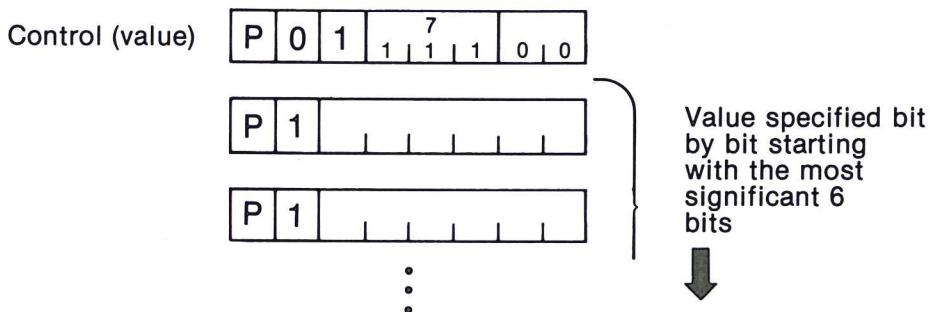
4.3 PDI CONTROL



The CONTROL opcode allows control over the states of the terminal and over the form of interpretation of the drawing opcode attributes such as line texture, crosshatching, etc. There are four forms of the CONTROL opcode distinguished by the facilities bits. These are presented below along with the facility bit combinations used to identify the form.

Facilities bit	Form	Description
00	Value	describe the colour or grey scale value
01	Status	provides extension to a field of sub-commands called Status Command
10	Reserved	reserved for future control commands for manipulating a displayed picture
11	Private	reserved for use by terminal manufacturers to implement private non-standard functions

The VALUE form of the CONTROL opcode defines the value (colour of the grey scale) accessed by subsequent drawing opcodes. For example, a CONTROL (VALUE) opcode could be transmitted to define the colour red, and then a RECTANGLE opcode transmitted to draw a rectangle. The rectangle would then be coloured red. The general CONTROL (VALUE) opcode is illustrated below:

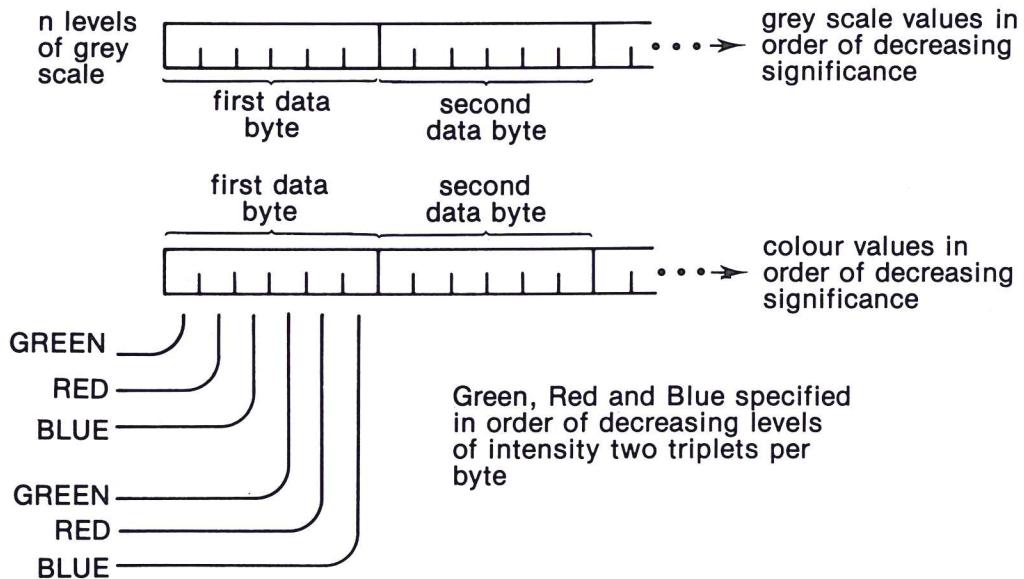


Terminals vary in the number of colours or levels of grey scale they support. The minimum is a system with merely one level to display only black and white monochrome. A typical system supports eight shades of grey scale, the eight colours Black, Blue, Red, Magenta, Green, Yellow, Cyan, White, blinking on White and a transparent mode to allow captioning of regular television pictures.

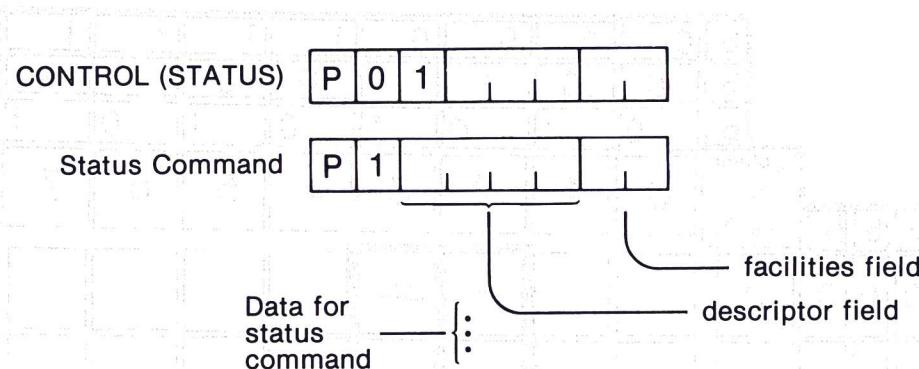
The control information which specifies whether a CONTROL (VALUE) opcode contains colour or grey scale data or whether blinking or transparent mode is invoked is specified in a Status Command. The CONTROL (VALUE) form merely interprets the associated data bytes as levels of grey scale of decreasing significance or of triplets of values for the three primary colours GREEN, RED and BLUE also of decreasing significance.

Forward and backward compatibility are maintained by specifying values in order of decreasing significance. This allows terminals which are not equipped to display many levels of colour or grey scale to simply truncate the levels of colour or grey scale to simply truncate the command data to the number of bits which can be handled. On the other hand, if a colour or grey scale is only specified to a few levels the resulting picture will be the same on both low and high resolution terminals. The primary colours are described in the order GREEN, RED, BLUE because this is the decreasing order of intensity of the three colours so that a colour image would appear as a meaningful grey scale picture on a monochrome terminal.

The bit assignments of CONTROL (VALUE) opcode are indicated below:



The CONTROL (STATUS) opcode defines the status of the drawing environment. It achieves this by extending into a field of sub-commands called Status Commands which occupy code positions in columns 4, 5, 6 and 7 of the 7-bit code table. The Status Commands are one or more byte sequences that immediately follows the CONTROL (STATUS) opcode byte. The format of this byte is subdelimited containing a four bit description field and a two bit facilities field.



The detailed meaning of each of the Status Command is presented herein.

The CONTROL (RESERVED) opcode has no action for the field trial and any data following it is ignored up to another opcode or SO, SI, SS(2) or ESC character. This opcode format is reserved for future use for manipulation of a displayed picture. Control commands in this form would be called Picture Manipulation Instructions (PMI) and would permit selective erasure, update and modification of the displayed picture.

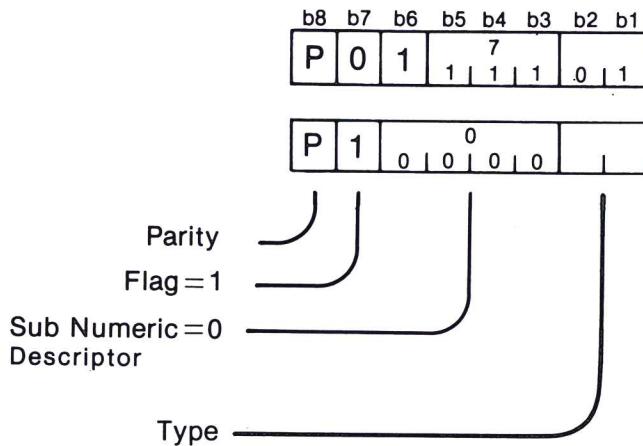
The CONTROL (PRIVATE) opcode is left up to the terminal manufacturer. He may use this command to implement any special control sequence which applies only to one particular version of a Videotex terminal, such as control of a hardcopy device. It must be remembered that because the CONTROL (PRIVATE) opcodes are not defined in the same way for each terminal they should only be used with special agreement between the host data supplier and the terminal data user.

A summary of all the opcodes which occupy code positions in columns 2 and 3 of the 7-bit code table are given in Figure 5 and the control codes are given in Figure 6.

b_5	0	0	0	0	1	1	1	1
b_6	0	0	1	1	0	0	1	1
b_5	0	1	0	1	0	1	0	1
	column	0	1	2	3	4	5	6
	row	0	1	2	3	4	5	7
b_4	b_3	b_2	b_1					
0	0	0	0	0	SPARE	RECT (Outlined)		
0	0	0	1	1	SPARE	RECT (Filled)		
0	0	1	0	2	SPARE	SET & RECT (Outlined)		
0	0	1	1	3	SPARE	SET & RECT (Filled)		
0	1	0	0	4	POINT SET (Absolute)	POLY (Outlined)		
0	1	0	1	5	POINT SET (Relative)	POLY (Filled)		
0	1	1	0	6	POINT (Absolute)	SET & POLY (Outlined)		
0	1	1	1	7	POINT (Relative)	SET & POLY (Filled)		
1	0	0	0	8	LINE (Absolute)	BIT (Point by Point)		
1	0	0	1	9	LINE (Relative)	BIT (Runlength)		
1	0	1	0	10	SET & LINE (Absolute)	SET & BIT (Point by Point)		
1	0	1	1	11	SET & LINE (Relative)	SET & BIT (Runlength)		
1	1	0	0	12	ARC (Outlined)	CONTROL (VALUE)		
1	1	0	1	13	ARC (Filled)	CONTROL (STATUS)		
1	1	1	0	14	SET & ARC (Outlined)	CONTROL (RESERVE)		
1	1	1	1	15	SET & ARC (Filled)	CONTROL (PRIVATE)		

Figure 5. Summary of OPCODES

4.3.0 CLEAR



The CLEAR Status Command provides general control over all states of the terminal by permitting general erasure of the entire display screen and also control to reset all the other states of the terminal to their initial conditions.

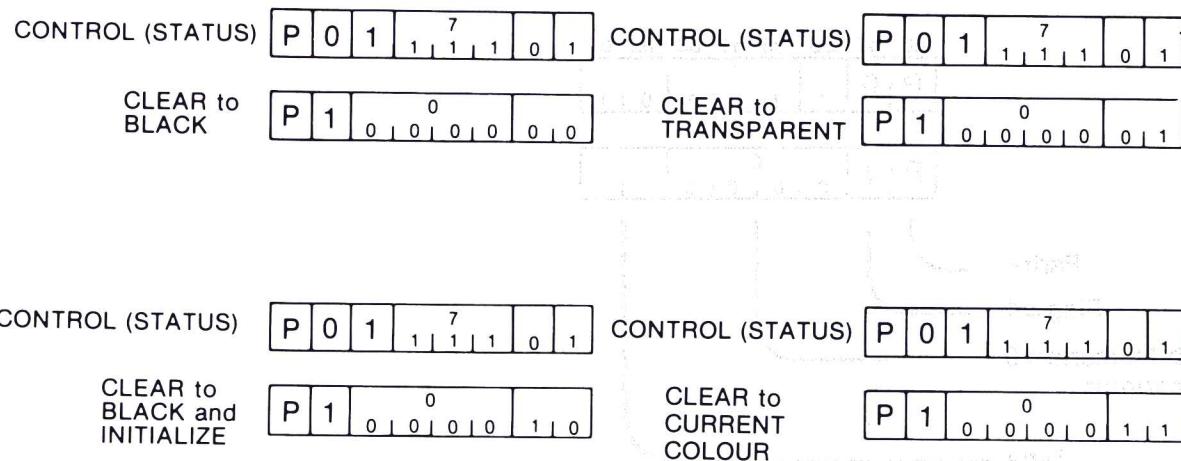
Facilities bits b1 and b2 are used to indicate what colour is to be loaded into the display memory upon a CLEAR command. As is discussed in more detail later, the TRANSPARENT level is used to effect the mixing of PDI images and conventional television pictures. When the terminal is cleared to a transparent state, the entire conventional television picture becomes visible. This state defaults to BLACK for a background if the terminal apparatus is not configured to mix PDI and television images.

An initialization form of the clear command returns the terminal to its initial default state; that is the terminal is in SI mode with all the status conditions reset to their default state. The initialization operation does not affect the input buffer of command yet to be processed. The non-initialization form of the CLEAR Status Command only erases the screen and resets the current drawing position to the "home" position as described for the SI mode. The "home" position of the current drawing position is at the left margin and one character height below the top, or in other words, the position before printing the first character. This is also considered the default location of the current drawing position.

OPERATION

- The CONTROL (STATUS) opcode is recognized and the byte that follows indicates the type of the Status Command.
- The CLEAR Status Command causes a general ERASE of the display screen. If facilities bits b2, b1 equal 0,0, this erase is to BLACK, but if facilities bit b2, b1 equal 0,1, the erase is the TRANSPARENT level. The current drawing position is set to the "home" character position.
- If any additional data bytes follow a CLEAR Status Command, they are ignored.

The four forms are:



4.3.1 DOMAIN

The DOMAIN Status Command permits the number domain for coordinate or relative displacement to be specified. As discussed previously, x,y positions or other numeric coordinate data associated with a PDI opcode are normally specified to 9 bits of accuracy encoded into three data bytes, but this encoding may be varied to 4, 5 or even 6 bytes of data. The DOMAIN Status Command defines the number of bytes specifying the number domain and consists of bits b2 and b1 of the DOMAIN code as illustrated below.

Parity

Flag=1

Sub Numeric=1

Descriptor

Domain

Code

The DOMAIN Status Command permits the number domain for coordinate or relative displacements to be specified. As discussed previously, x,y positions or other numeric coordinate data associated with a PDI opcode are normally specified to 9 bits of accuracy encoded into three data bytes, but this encoding may be varied to 4, 5 or even 6 bytes of data. The DOMAIN Status Command defines the number of bytes specifying the number domain and consists of bits b2 and b1 of the DOMAIN code as illustrated below.

DOMAIN code	bits in number	number of bytes	encoded in bytes
* 00	9	3	00000000
01	12	4	000000000000
10	15	5	00000000000000
11	18	6	0000000000000000

* default setting for both command and PDI & PDI command related to resolution via

OPERATION

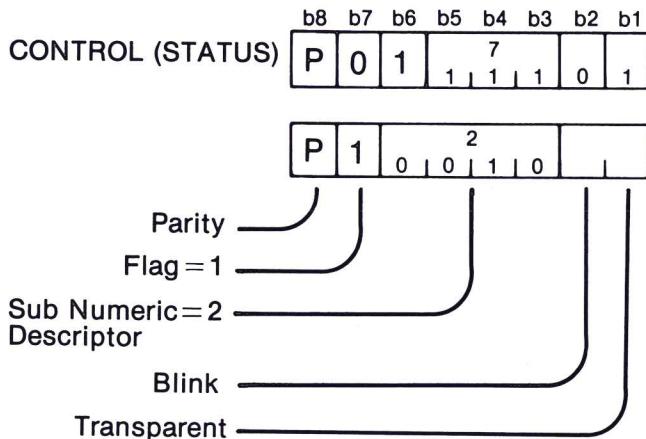
NUMBER DOMAIN STATUS

- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.
- The DOMAIN Status Command word is interpreted and the DOMAIN code is set up as the Number Domain Status.
- If any additional data bytes follow a DOMAIN control command, they are ignored.

The four formats are:

CONTROL STATUS <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table> DOMAIN CODE <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table> <p>(3 bytes to be used to specify an x, y coordinate, the default)</p>	P	0	1	1	1	1	1	0	1	P	1	0	0	0	0	1	1	0	0	CONTROL (STATUS) <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table> DOMAIN CODE <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table> <p>(4 bytes to be used to specify an x, y coordinate, the default)</p>	P	0	1	1	1	1	1	0	1	P	1	0	0	0	0	1	1	0	1
P	0	1	1	1	1	1	0	1																															
P	1	0	0	0	0	1	1	0	0																														
P	0	1	1	1	1	1	0	1																															
P	1	0	0	0	0	1	1	0	1																														
CONTROL (STATUS) <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table> DOMAIN CODE <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table> <p>(5 bytes to be used to specify an x, y coordinate, the default)</p>	P	0	1	1	1	1	1	0	1	P	1	0	0	0	0	1	1	1	0	CONTROL (STATUS) <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table> DOMAIN CODE <table border="1" style="margin-left: 10px; border-collapse: collapse;"> <tr><td>P</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table> <p>(6 bytes to be used to specify an x, y coordinate, the default)</p>	P	0	1	1	1	1	1	0	1	P	1	0	0	0	0	1	1	1	1
P	0	1	1	1	1	1	0	1																															
P	1	0	0	0	0	1	1	1	0																														
P	0	1	1	1	1	1	0	1																															
P	1	0	0	0	0	1	1	1	1																														

4.3.2 DRAWING CONTROL



There are two terminal status modes controlled by the DRAWING CONTROL Status Command. They are BLINK and TRANSPARENT status which are taken directly from the facilities bits b2 and b1. Both of these modes may optionally be implemented or partially implemented by terminal manufacturers. If these modes are not implemented then this DRAWING CONTROL Command has no effect.

The BLINK mode of the terminal allows particular objects drawn on the display screen to be flashed in a repetitive manner for emphasis. Any LINE, RECTANGLE or other item specified by a PDI opcode can be blinked by setting the blink status flag before drawing the object. The end of BLINK status is indicated by another DRAWING CONTROL Status Command in which the BLINK status bit is cleared. In general an object of any colour or grey scale level may be blinked, but in some implementations, blinking may be restricted to white, so that any object which is specified to blink will appear in the colour WHITE.

The TRANSPARENT mode of the terminal is used to mix graphics with conventional television signals for the purpose of captioning, annotating or other special effects. The PDI mode of the terminal can be considered as overwriting conventional television pictures. One manner of using transparent mode is to CLEAR the display screen to transparent and then to write graphics areas, text, etc. over the conventional television picture. Another way of using transparent mode is to use RECTANGLE, POLYGON, etc. to draw "transparent" shapes, that is, to draw shapes in which a background conventional television picture will show through. The DRAWING CONTROL Status Command may be used to set the TRANSPARENT status flag. When the TRANSPARENT status flag is set all drawing commands write in a "transparent" colour. The end of TRANSPARENT status is indicated by another DRAWING CONTROL Status Command in which the TRANSPARENT bit is cleared. If the TRANSPARENT feature has not been implemented on a particular terminal, it has no effect. The order of precedence is TRANSPARENT, BLINK and then COLOUR.

OPERATION

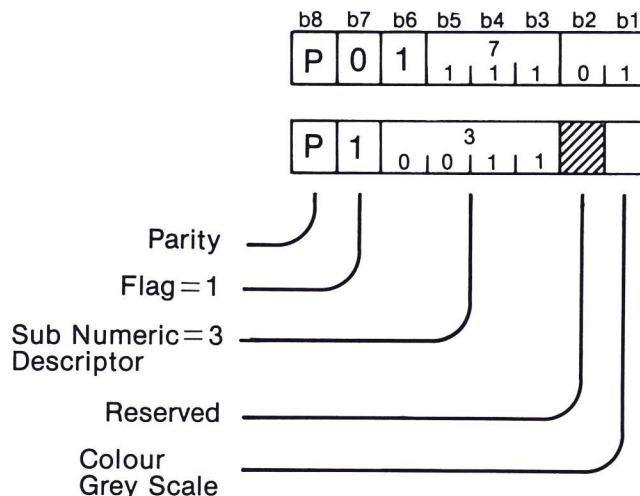
- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.

- The DRAWING CONTROL Status Command sets the BLINK and TRANSPARENT status flags with the values of the two facilities bits b2 and b1. Facilities bit values of 1 are interpreted as being the ON states for the BLINK and TRANSPARENT modes.
- If any additional data bytes follow a DRAWING CONTROL Status Command they are ignored.

The four forms are:

CONTROL (STATUS) <table border="1"><tr><td>P</td><td>0</td><td>1</td><td></td><td>7</td><td></td><td></td><td>0</td><td>1</td></tr><tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td></td><td>0</td><td></td></tr></table>	P	0	1		7			0	1				1	1	1		0		CONTROL (STATUS) <table border="1"><tr><td>P</td><td>0</td><td>1</td><td></td><td>7</td><td></td><td></td><td>0</td><td>1</td></tr><tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td></td><td>0</td><td></td></tr></table>	P	0	1		7			0	1				1	1	1		0	
P	0	1		7			0	1																													
			1	1	1		0																														
P	0	1		7			0	1																													
			1	1	1		0																														
DRAWING CONTROL <table border="1"><tr><td>P</td><td>1</td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td></td></tr></table> (BLINK OFF TRANSPARENT OFF)	P	1		2							0	0	1	1	0	0	0		DRAWING CONTROL <table border="1"><tr><td>P</td><td>1</td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td></td></tr></table> (BLINK OFF TRANSPARENT ON)	P	1		2							0	0	1	1	0	0	1	
P	1		2																																		
	0	0	1	1	0	0	0																														
P	1		2																																		
	0	0	1	1	0	0	1																														
CONTROL (STATUS) <table border="1"><tr><td>P</td><td>0</td><td>1</td><td></td><td>7</td><td></td><td></td><td>0</td><td>1</td></tr><tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td></td><td>0</td><td></td></tr></table>	P	0	1		7			0	1				1	1	1		0		CONTROL (STATUS) <table border="1"><tr><td>P</td><td>0</td><td>1</td><td></td><td>7</td><td></td><td></td><td>0</td><td>1</td></tr><tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td></td><td>0</td><td></td></tr></table>	P	0	1		7			0	1				1	1	1		0	
P	0	1		7			0	1																													
			1	1	1		0																														
P	0	1		7			0	1																													
			1	1	1		0																														
DRAWING CONTROL <table border="1"><tr><td>P</td><td>1</td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td></td></tr></table> (BLINK ON TRANSPARENT OFF)	P	1		2							0	0	1	1	0	1	0		DRAWING CONTROL <table border="1"><tr><td>P</td><td>1</td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td></td></tr></table> (BLINK ON TRANSPARENT ON)	P	1		2							0	0	1	1	0	1	1	
P	1		2																																		
	0	0	1	1	0	1	0																														
P	1		2																																		
	0	0	1	1	0	1	1																														
TRANSPARENT supersedes BLINK when implemented																																					

4.3.3 TONAL CONTROL



The Control (VALUE) opcode defines the colour or greyscale value accessed by subsequent drawing operations, as described in Section 4.3. This value may be used to represent either colour or greyscale value for a

particular picture element as defined by the TONAL CONTROL status flag. The TONAL CONTROL is initially 0 indicating that all subsequent drawing operations would produce coloured lines, rectangles, text, etc. If the TONAL CONTROL flag is set to 1, all subsequent drawing operations would produce picture drawings in which the value would be interpreted as a greyscale level. Both coloured and greyscale graphical drawing entities may co-exist in the same picture. The drawing operations to create the coloured and greyscale picture are separated by TONAL CONTROL Status Commands.

The facility bit b2 is reserved for possible future use with colour reference tables.

When a TONAL CONTROL Status Command is issued the low order facilities bit b1 is loaded directly into the status flag.

OPERATION

- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.
- Facilities bit b1 of the second command byte is loaded directly into the TONAL CONTROL status flag.
- If any additional data bytes follow, they are ignored.

The four forms are:

CONTROL (STATUS)

P	0	1		7			
			1	1	1	0	1

TONAL
CONTROL
(colour)

P	1		3				
	0	0	1	1	0	0	0

CONTROL (STATUS)

P	0	1		7			
			1	1	1	0	1

TONAL
CONTROL
(colour)

P	1		3				
	0	0	1	1	1	1	0

CONTROL (STATUS)

P	0	1		7			
			1	1	1	0	1

TONAL
CONTROL
(grey
scale)

P	1		3				
	0	0	1	1	0	1	0

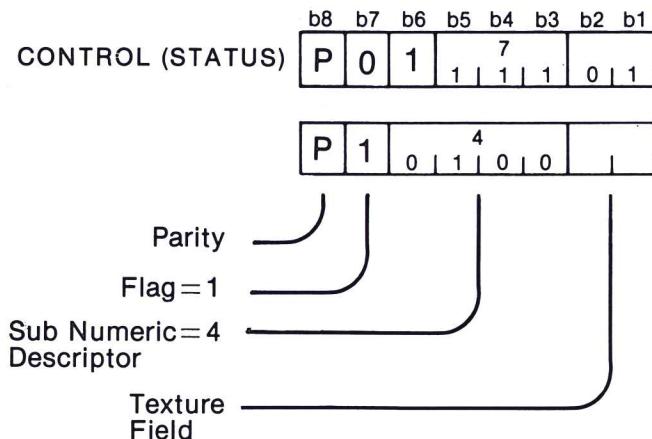
CONTROL (STATUS)

P	0	1		7			
			1	1	1	0	1

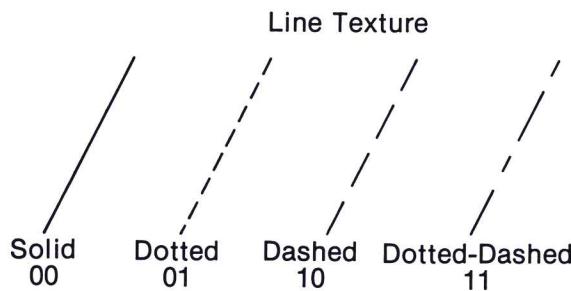
TONAL
CONTROL
(grey
scale)

P	1		3				
	0	0	1	1	1	1	1

4.3.4 LINE CONTROL



Line texture controls the form in which LINES or outlines for a RECTANGLE, POLYGON or ARC are drawn. The following line textures are available.



The LINE CONTROL status is defined in a two bit texture field consisting of facilities bits b2 and b1 from the LINE CONTROL Status Command. The default line texture is Ø indicating SOLID. If the LINE CONTROL Status Command is issued, subsequent lines are drawn with the specified line texture. The subsequent byte is reserved for the future to specify line thickness. The line texture pattern is referenced to the absolute coordinate grid of the display screen so that the texture pattern aligns between drawing commands.

OPERATION

- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.
- Facilities bits b2 and b1 are interpreted as a texture control field and are loaded directly into the line texture status register.
- If any additional data bytes follow, they are ignored.

The four forms are:

CONTROL (STATUS)

P	0	1	7		0	1
		1	1	1	0	

 CONTROL (STATUS)

P	0	1	7		0	1
		1	1	1	0	

LINE
CONTROL
(SOLID)

P	1	4			0	0
	0	1	0	0	0	

LINE
CONTROL
(DOTTED)

P	1	4			0	1
	0	1	0	0	0	1

CONTROL (STATUS)

P	0	1	7		0	1
		1	1	1	0	

 CONTROL (STATUS)

P	0	1	7		0	1
		1	1	1	0	

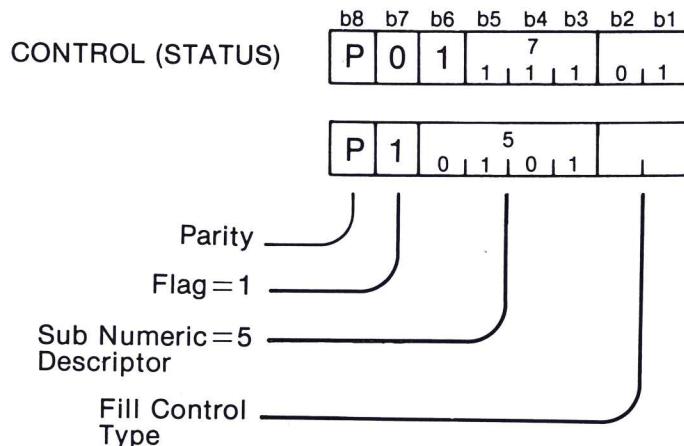
LINE
CONTROL
(DASHED)

P	1	4			1	0
	0	1	0	0	1	0

LINE
CONTROL
(DOT-DASHED)

P	1	4			1	1
	0	1	0	0	1	1

4.3.5 FILL CONTROL



The FILL CONTROL Status Command provides the means for establishing the fill pattern (crosshatching), ARC fill form and the highlight mode. The two facilities bits b2 and b1 are used to indicate the type of the FILL CONTROL Status Command as illustrated below.

- 00 – fill (no highlight)
- 01 – fill and highlight – same colour
- 10 – fill and highlight – black
- 11 – reserved for future bit value mask

The FILL CONTROL Status Command allows the manner in which areas are filled to be controlled. An associated numeric data item of dx and dy specifies the spacing of the lines used to fill an area. Lines spaced across an area will produce a crosshatched effect which may be used in much the same way as line texture for line drawings. Line texture may also be used in conjunction with crosshatching so that an area may be filled with DOTTED, DASHED or DOT-DASHED lines. In this way line texture also applies to area filling.

An additional group of data bytes is associated with a FILL PATTERN Status Command to specify the spacing of the crosshatching in terms of dx and dy. If both dx and dy equal \emptyset no crosshatching is performed and the area is entirely filled. If dx or dy equals \emptyset then crosshatching is performed in the direction for which a line spacing is specified. If a line space is specified for both directions then crosshatching in both x and y is performed. The meaning of the parameters of the fill command corresponds to the fraction of fill in that particular direction, that is:

\emptyset in the X or Y means do not fill in that direction.

1 in the X or Y means fill to full density in that direction.

A fraction specified in the X or Y direction means crosshatch fill to that density in that direction, i.e., .5 in X means fill every second pixel element in X in a crosshatching manner by means of vertical lines.

1 in both X and Y means fill to full density where the direction is up to the manufacturer.

\emptyset in both directions means the same as 1 in both X and Y.

Negative values for fill crosshatch density are used to indicate angled fill; that is, an area may be filled by a crosshatch pattern which is angled at 45 (slope 1) or at 135 (slope -1). If an x density specification is positive, the area will be filled by vertical line spaces such as to provide a fill density as specified for X. If X density specification is negative, then the area will be filled by vertical lines sloped 45 (in a positive mathematical sense) i.e., at slope of -1. Similarly for Y density.

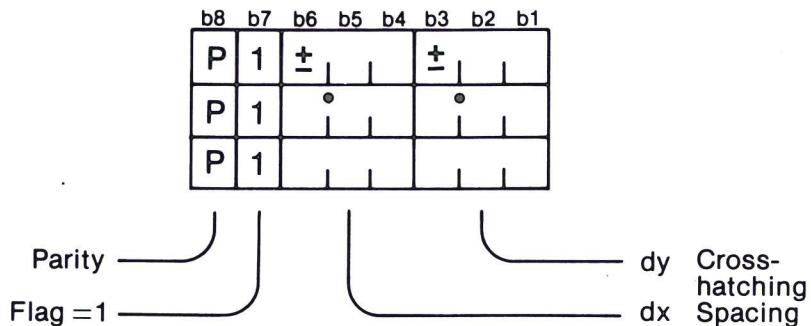
The forms of crosshatch filling direction are:

+ X		vertical lines spaced to the density in X
- X	\\ \\	sloped lines (slope -1) to the density specified in X
+ Y	== ==	horizontal lines spaced to the density specified in Y
- Y	// //	sloped lines (slope 1) to the density specified in Y

It is expected that a manufacturer who implements crosshatching of filled areas at all should implement at least four distinct crosshatch densities.

The direction of fill determines the orientation of crosshatch patterns and line texture, however, it does not necessarily define the order in which a filled area is drawn. This freedom is left to the manufacturer as long as the resultant picture is as described.

For both crosshatching and the line texture patterns the orientation of the pattern is fixed with respect to the absolute screen coordinates. For example, crosshatching in the x direction is referenced to the left hand margin of the screen. This means that two areas filled with the same crosshatching spacing will have their patterns aligned and adjacent areas will appear to be filled as if they were a single area.



Crosshatch density is specified in the same data format as coordinate specifications in terms of a binary fraction from \emptyset to 1 and using the same DOMAIN specification. This crosshatch density specification data must be included with a FILL Control Command.

The Highlight mode of the FILL CONTROL Status Command permits a POLYGON, an ARC or a RECTANGLE to be outline highlighted as well as filled. When a POLYGON, RECTANGLE or ARC is filled, it may be outlined, either in the same colour or greyscale value as the fill in order to sharpen up the edges, or outlined in BLACK to add contrast to the edges.

The Reserved form of the FILL CONTROL Status Command is intended to control a bit value masked which may be used in the future to allow facsimile type pictures to build up a level at a time with the most significant bits specified first.

OPERATION

- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.
- The Status Command byte is interpreted to determine the type of FILL CONTROL command. The two low order facilities bits b2 and b1 are used as a code to indicate this.
- Subsequent data bytes contain the fill pattern density specification.
- If any additional data bytes follow, they are ignored.

The four forms are:

CONTROL STATUS

P	0	1	1	1	1	0	1
---	---	---	---	---	---	---	---

FILL CONTROL

P	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---

No Highlight
(Fill Pattern)

P	1		dx		dy		
P	1						
P	1						

CONTROL STATUS

P	0	1	1	1	1	0	1
---	---	---	---	---	---	---	---

FILL CONTROL

P	1	0	1	5	0	1	0
---	---	---	---	---	---	---	---

Highlight in colour
(Fill Pattern)

P	1		dx		dy		
P	1						
P	1						

CONTROL STATUS

P	0	1	1	1	1	0	1
---	---	---	---	---	---	---	---

FILL CONTROL

P	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Highlight in BLACK
(Fill Pattern)

P	1		dx		dy		
P	1						
P	1						

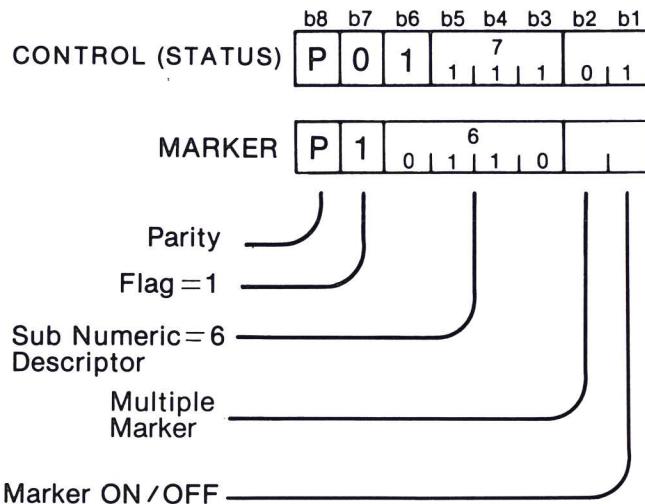
CONTROL STATUS

P	0	1	1	1	1	0	1
---	---	---	---	---	---	---	---

FILL CONTROL
(RESERVED)

P	1	0	1	5	0	1	1
---	---	---	---	---	---	---	---

4.3.6 MARKER



The MARKER is a small mark which may be displayed on the screen and moved about to point out a particular position or displayed items. The MARKER may be either turned ON or OFF, or may be positioned to any point on the screen. The MARKER appears to operate in an independent manner from other displayed material; that is, it superimposes in front of the display and when it is moved the image which was previously on the display at that point is re-generated. Manufacturers are free to implement many different styles for the marker, but it is recommended the marker shape somehow indicate its centre such as by a cross or a dot in the middle of a circle. Manufacturers may use different techniques to effect a marker. It is not guaranteed that the marker will remain visible if it is overwritten by a drawing command or that the background picture can be properly regenerated in such a case.

Bit b1 indicates whether the marker is to be set ON or OFF, and bit b2 is reserved to permit multiple markers. The marker position may be specified in an associated coordinate description. The default position of the marker is coordinate (\emptyset, \emptyset) .

The implementation of the marker is considered optional for a field trial terminal.

OPERATION

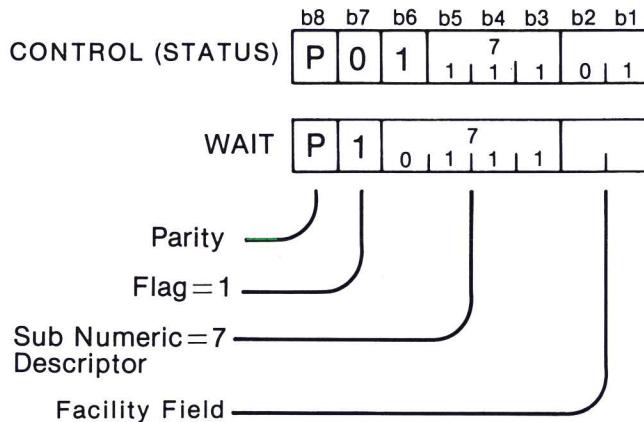
- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.
- If facilities bit b1 of the second command byte equals 1 the marker is enabled; however, if bit b1 equals \emptyset , the marker is set to its default off state.
- If facilities bit b2 equals \emptyset , the marker command refers to the current marker (default number \emptyset); however, if bit b2 equals 1, a byte is used to reference a specific marker (where multiple markers are supported).
- If any additional data bytes follow, they are ignored.

The four forms are:

CONTROL STATUS	<table border="1"><tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	P	0	1	1	1	1	0	1	CONTROL STATUS	<table border="1"><tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	P	0	1	1	1	1	0	1																																
P	0	1	1	1	1	0	1																																												
P	0	1	1	1	1	0	1																																												
CURRENT MARKER	<table border="1"><tr><td>P</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	P	1	0	1	1	1	0	0	0	CURRENT MARKER	<table border="1"><tr><td>P</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>	P	1	0	1	1	1	0	0	1																														
P	1	0	1	1	1	0	0	0																																											
P	1	0	1	1	1	0	0	1																																											
(marker off and a reposition of the Marker to x, y)	<table border="1"><tr><td>P</td><td>1</td><td>x</td><td></td><td>y</td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	P	1	x		y				P	1							P	1							(marker on and a reposition of the Marker to x, y)	<table border="1"><tr><td>P</td><td>1</td><td>x</td><td></td><td>y</td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr></table>	P	1	x		y				P	1							P	1						1
P	1	x		y																																															
P	1																																																		
P	1																																																		
P	1	x		y																																															
P	1																																																		
P	1						1																																												

CONTROL STATUS	<table border="1"><tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	P	0	1	1	1	1	0	1	CONTROL STATUS	<table border="1"><tr><td>P</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	P	0	1	1	1	1	0	1																																
P	0	1	1	1	1	0	1																																												
P	0	1	1	1	1	0	1																																												
MARKER	<table border="1"><tr><td>P</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	P	1	0	1	1	1	0	1	0	MARKER	<table border="1"><tr><td>P</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	P	1	0	1	1	1	0	1	1																														
P	1	0	1	1	1	0	1	0																																											
P	1	0	1	1	1	0	1	1																																											
MARKER NUMBER	<table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	P	1							MARKER NUMBER	<table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	P	1																																						
P	1																																																		
P	1																																																		
(marker off and a reposition of the Marker to x, y)	<table border="1"><tr><td>P</td><td>1</td><td>x</td><td></td><td>y</td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	P	1	x		y				P	1							P	1							(marker on and a reposition of the Marker to x, y)	<table border="1"><tr><td>P</td><td>1</td><td>x</td><td></td><td>y</td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <table border="1"><tr><td>P</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	P	1	x		y				P	1							P	1						
P	1	x		y																																															
P	1																																																		
P	1																																																		
P	1	x		y																																															
P	1																																																		
P	1																																																		

4.3.7 WAIT



The WAIT Status Command is a command with no result but which causes a delay in processing of a specific time. Time delays may be specified in terms of tenths of a second, in an associated data byte of six bits width which permits a maximum wait of 6.3 seconds, or in two associated data bytes which permits a maximum time delay of 6.8 minutes. The least significant data byte is specified first so that short delays may be specified with less coding by simply dropping the most significant data byte.

The activation of any interaction sequence on an interactive device terminates a WAIT command.

If facilities bit B1 is set then the WAIT is an infinite WAIT.

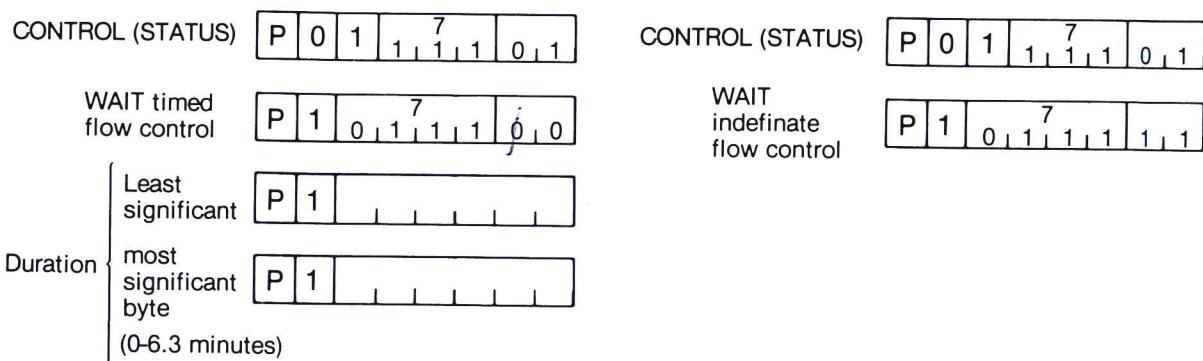
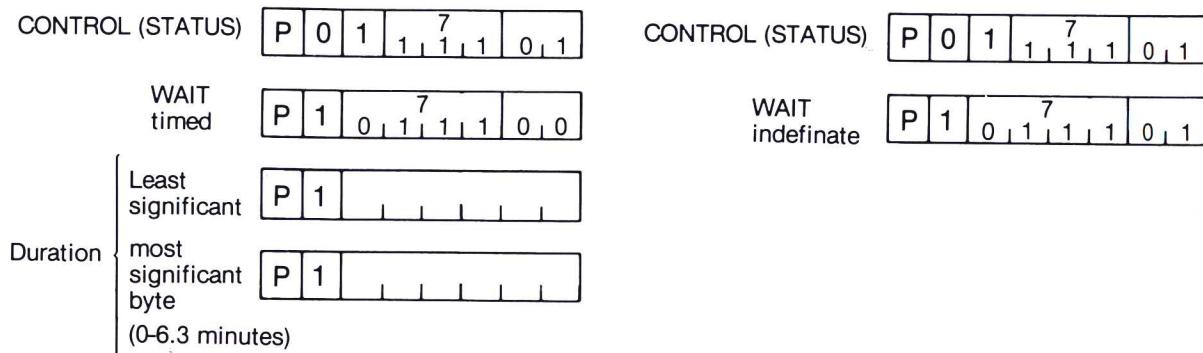
In some types of terminals and/or interactive scenarios the WAIT command may cause flow control to occur on the communications line. Such a WAIT command being processed by the terminal sends a DC3 (pause flow) control to the host computer so that subsequent data transmissions are halted and data does not accumulate in the terminal input buffer. The termination of this WAIT then sends a DC1 command to the host to resume data flow. This facility results in removing the majority of the effect of buffering data on the rate of display. In order to entirely remove this effect, it is necessary for the host data base computer to enter an idle state after sending a WAIT command, but this protocol does not depend upon a host computer implementing such an idle condition. Facilities bit b2 is used to indicate if flow control is to be used.

The WAIT command provides a CONCEAL/REVEAL capability which is useful in instructional and response situations. As an example of such a situation a page of questions could be displayed and then a WAIT issued followed by the answer being transmitted to a terminal. The answer would be in the input buffer of the terminal, and it would appear either after the established time or when the user interacted with his keypad or other interactive device. Multiple WAITS could be issued in this manner to handle a list of questions and the answers could be text or diagrams.

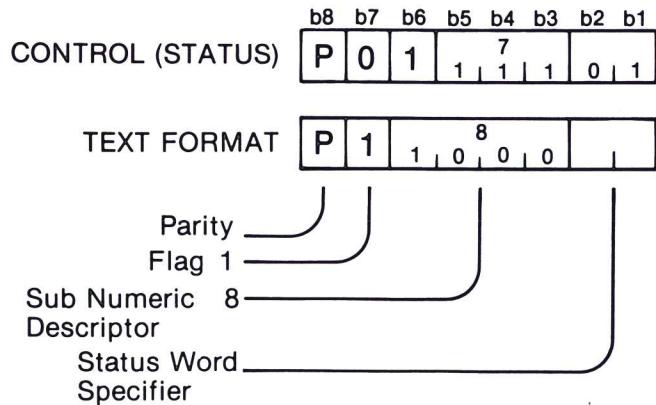
OPERATION

- The CONTROL (STATUS) opcode is recognized and the following bits indicate a STATUS command. The following byte indicates the type of the Status Command.
- If facilities bit b1 equals 0, a timed WAIT is indicated, otherwise if facilities bit b1 equals 1 and infinite WAIT is indicated.
- If facilities bit b2 equals 0, flow control is indicated, otherwise if facilities bit b2 equals 1, no flow control is used.
- The second command byte indicates that a wait operation is to be performed.
- The next associated data byte indicates the least significant six bits of the time duration and optionally the next data byte contains the most significant six bits of the time duration.
- If any further data bytes follow, they are ignored.

The four forms are:



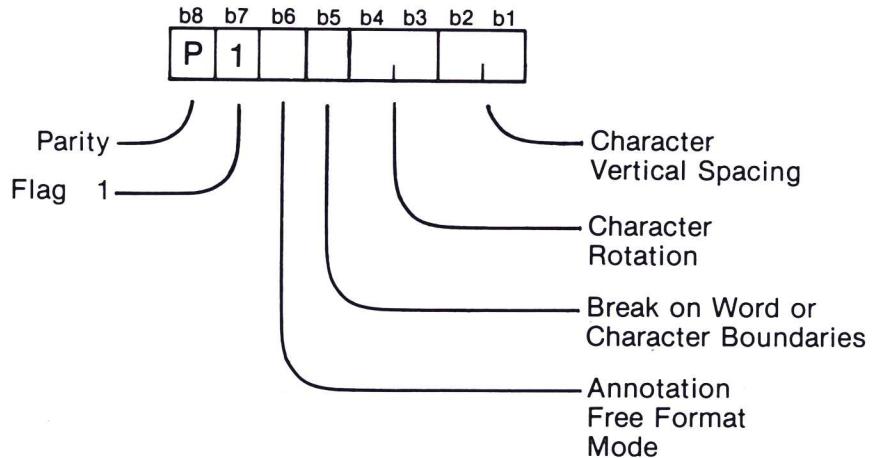
4.3.8 TEXT FORMAT



The TEXT FORMAT Status Command describes the manner in which characters are presented. The two low order bits b2 and b1 make up a Status Word Specifier which describes the particular function of the command. These codes are:

- 00 – CHARACTER FORMAT
- 01 – CHARACTER SIZE
- 10 – CHARACTER SET
- 11 – TAB POSITIONS

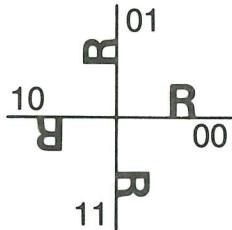
00 – CHARACTER FORMAT entails the control of several different options in the manner in which characters are displayed. The associated data byte following the TEXT FORMAT Status Command has the following form:



As described previously, there are two modes of presenting the text information, Format \emptyset , Free Format mode in which the size and format of character presentation is determined in the terminal, and format 1, used for annotating diagrams and graphics in which the character size is specified. Bit b6 indicates the format, which is initially \emptyset (Format \emptyset), the system default. This means that in the default state a videotex terminal operates like a conventional business terminal.

In format \emptyset text mode, character strings may automatically wraparound when the right margin of the screen is reached. There are two forms of this wraparound. In the first or default mode (bit b5 = \emptyset) character strings are broken on a character by character basis. If a character to print is beyond the margin it is printed on the next line. In the second mode (bit b5 = 1) character strings are broken on a word basis. Character strings are buffered until a space (the word separator) or a punctuation marks such as , . : ; -) = * / ? ! + is encountered, and then the "word" is printed in its entirety on one line. If the word is longer than a line, i.e., if the buffer fills to greater than the number of characters in a line, then the buffer is output as a line broken arbitrarily within the word. These two modes are necessary, \emptyset to emulate a terminal mode and 1 to allow word oriented text to be presented in a word oriented format.

Character Rotation is optionally available and at that only on format 1 character presentation in the TEXT mode. The automatic wraparound and scrolling available in format \emptyset precludes character rotation. Character rotation is specified in a 2 bit field which allows for 4 directions of rotation, as illustrated below:



Rotated strings of characters proceed in the direction of the rotation, for example, a character string of rotation $\emptyset 1$ increments character by character in the vertical direction upwards.

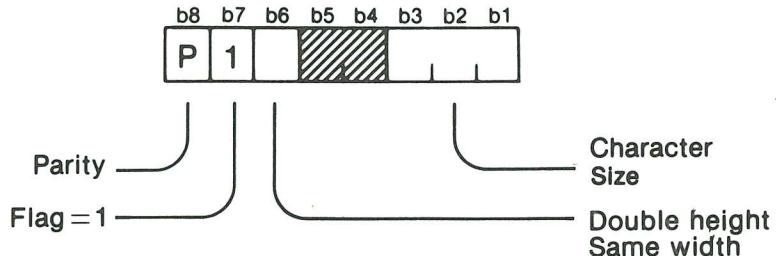
Rotated characters are optional and may not be implemented on some terminals, while other terminals which cannot handle rotated character may print characters with the normal orientation but present the string of characters in the altered direction. Regardless of whether rotated characters are implemented, partially implemented or not implemented, the control characters CR, LF, HT, VT, BS will have their normal non rotation orientation meanings, that is, a carriage return always returns to the X position SET in the last set command.

VT always moves in the + Y direction
 BS always moves in the - X direction
 HT always moves in the + X direction
 LF always moves in the - Y direction

It is not good practice to use these format effector characters with rotated characters.

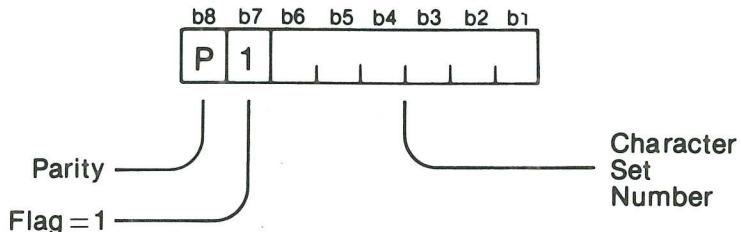
Vertical character spacing is like that of a typewriter and may be set at 1, 1.5, 2, 2.5 spaces per row of characters. This reduces the number of rows of text which may be displayed on the screen but increases legibility in many cases. In languages where accents must be put over capital letters such as Ñ in Spanish, it is desirable to use 1.5 spacing as a minimum to prevent the accent from interfering with the tail of descending small characters such as "g" on the row above. The four spacing codes are taken from the low order bits b2 and b1. The default is Ø which is single spaced. The vertical spacing capability is available for format 1, annotation text. It is left to the manufacturer as to whether it applies to format Ø text.

CHARACTER SIZE may be explicitly specified for character presentation form 1 for annotation, and is described by a number from Ø to 7 as indicated previously. This number is supplied in data word Ø1. The default is size 3.

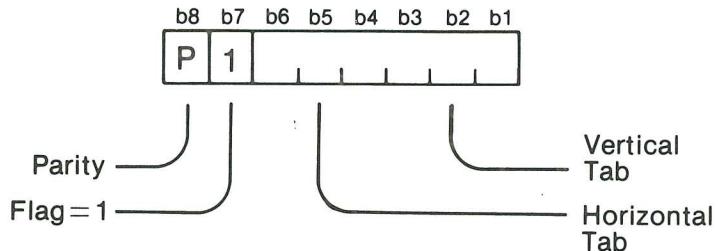


Any character size may be specified as being double height and of the same width described in the size field.

As mentioned previously, CHARACTER SETS are described by a status word containing a set number from Ø to 63. Set Ø is reserved for a user defined set which can be loaded into the terminal at any time. The default set is number 1. For the field trial, only sets 1 and 2 will be implemented.



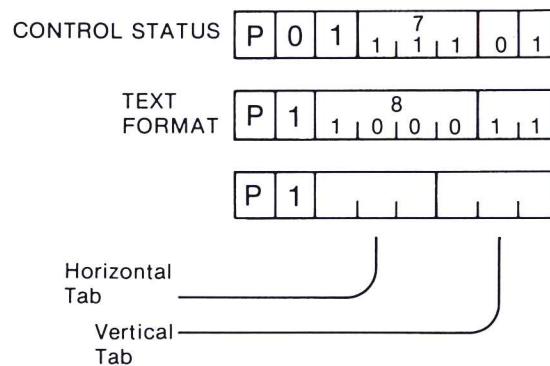
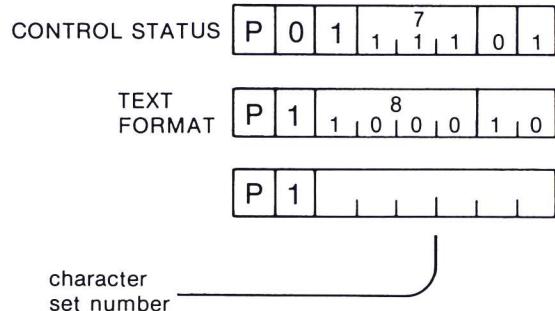
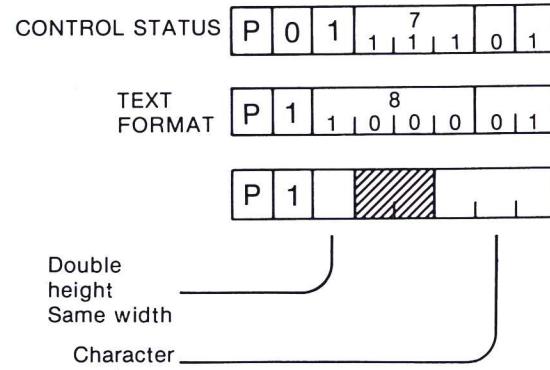
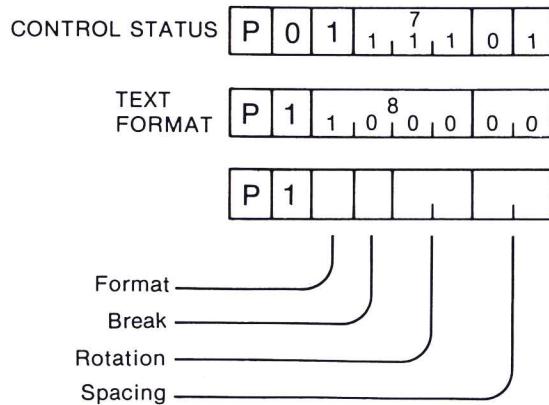
TAB POSITIONS are also controlled by the TEXT FORMAT Status Command. The default vertical and horizontal tab positions are 1 character space wide or high. This may be re-specified as being anywhere from 1 to 8 spaces. Since international agreement has not been achieved for the tab characters in video-tex systems, it is suggested that these characters be used sparingly in order to facilitate international interworking at the text level.



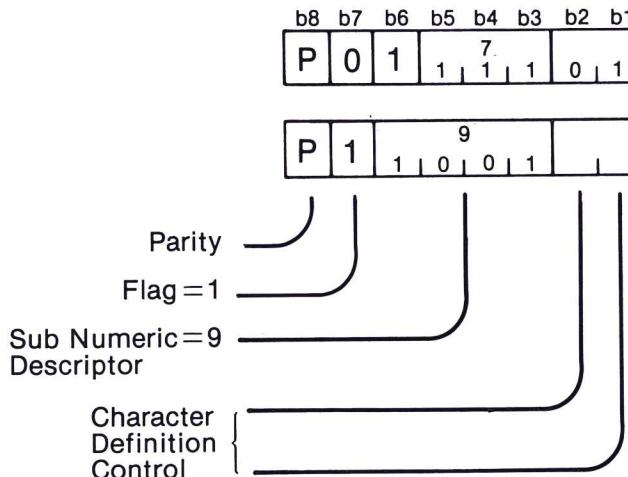
OPERATION

- The CONTROL (STATUS) opcode is recognized and the following byte indicates the type of the Status Command.
- The second command byte indicates that the instruction is a TEXT FORMAT Status Command. The two low order bits indicate the forms of the command.
- If bits b2 and b1 = $\emptyset\emptyset$, then the instruction is a CHARACTER FORMAT.
- If bits b2 and b1 = \emptyset , then the instruction is a CHARACTER SIZE.
- If bits b2 and b1 = $1\emptyset$, then the instruction is a CHARACTER SET.
- If bits b2 and b1 = 11 , then the instruction is a TAB POSITION.
- If any further data bytes follow they are ignored.

The four forms are:



4.3.9 CHARACTER DEFINITION



Character set \emptyset is a user definable character or symbol set which allows the user to describe a character of his own design. Mathematical, other special characters or character sets not built into the terminal may be defined in this manner.

There are several methods of encoding the character forms for communication to the terminal, and research is ongoing to determine the best format. At present, the CHARACTER DEFINITION control command performs no operation, but is reserved for the implementation of definable character sets.

4.4 ERRORS

Parity is used on all bytes of data transmitted in the PDI code set, both to be compatible with standard computer communications facilities and for basic error detection. On some communication facilities other communications error detecting codes may be used in association with the parity checking for higher reliability, but the only error detection performed in the terminal is for parity errors.

The parity form to be used in the Canadian Videotex field trials is odd parity. Terminal manufacturers may wish to build terminals with an odd/even/no parity switch or with autoparity software in order to address a wider market than just these field trials, but odd parity should be the default.

A parity error indicates that a specific byte of data is in error. This may or may not be serious depending upon the state of the terminal. Most commands to the terminal involve many bytes of coding and so it is necessary to reject the whole command if a parity error is detected. In other instances such as the display of text, it is only necessary to drop single characters in error.

The following is an outline of the minimum error handling which should be performed by a PDI terminal.

- For all errors in the opcode of a command, ignore the entire command until the next opcode. Indicate this error by an error marker.
- For errors in the body of a command, do not perform the action for which the erroneous data occurred and ignore the rest of the command until the next opcode. Indicate this error by an error marker.
- For errors in TEXT command on a single character, drop the erroneous character and display a filled in character (pillow) in its place.
- For errors in a BIT command, drop the erroneous value and display an arbitrary value of pixel or pixels in its place, i.e., BLACK, WHITE, preceding value or average value at the manufacturer's discretion).

CØ CONTROL codes do not occur in the bit patterns of any PDI command except for the carriage control codes which are used with TEXT. If any control codes are in error, i.e., out as sequence, then they are ignored.

It is recommended that a terminal indicate to the user in some way that an uncorrected error has been detected and that therefore the picture which has been displayed is in some way corrupted. The form of the error indicator will be left to the manufacturer, but it is suggested that a flashing white X be placed in the lower left hand corner of the screen.

5. RESPONSE CODES

The Picture Description Instructions presented so far through this document have described the manner in which pictures are specified in order to be drawn. A Videotex terminal utilizes communication codes in the direction from the terminal to the host computer to encode the interactions and responses of the user. These user responses occur independently of the Picture Description Instructions and are coded in a distinctly different manner.

As mentioned in Section 1.2 there are distinct classes of interaction into which the various types of interactive devices may be placed.

Sophisticated graphical devices may be used with future Videotex terminals for information preparation purposes or interactive communications. The functions may be realized through various hardware implementations and, in fact, the same device may perform two or more different functions. For example, a joystick (Control arm) could be used to control the position of a marker in picture creation, or to indicate objects already displayed on the screen. That is, it can be used both as a LOCATOR and as an IDENTIFIER. It is the function being performed which is important and must be communicated from the terminal for all interactions. Control codes to communicate these functions are needed but should not be related to the hardware device performing the function.

b ₇	0	0	0	0	1	1	1	1
b ₆	0	0	1	1	0	0	1	1
b ₅	0	1	0	1	0	1	0	1
	0	1	2	3	4	5	6	7
b ₄	b ₃	b ₂	b ₁	b ₀				
0	0	0	0	0	CLEAR (To Black)	LINE CONTROL (Solid)	TEXT FORMAT (Character Format)	
0	0	0	1	1	CLEAR (To Transparent)	LINE CONTROL (Dotted)	TEXT FORMAT (Character Size)	
0	0	1	0	2	CLEAR (To Black & Initialize)	LINE CONTROL (Dashed)	TEXT FORMAT (Character Set)	
0	0	1	1	3	CLEAR (current colour)	LINE CONTROL (Dot-Dashed)	TEXT FORMAT (Line position)	
0	1	0	0	4	DOMAIN (9 Bits)	FILL (Pattern)	CHAR DEF (Reserved)	
0	1	0	1	5	DOMAIN (12 Bits)	FILL highlight colour	CHAR DEF (Reserved)	
0	1	1	0	6	DOMAIN (15 Bits)	FILL highlight black	CHAR DEF (Reserved)	
0	1	1	1	7	DOMAIN (18 Bits)	FILL (Reserved)	CHAR DEF (Reserved)	
1	0	0	0	8	DRAWING (Blink OFF Transp OFF)	MARKER OFF NO Reposition		
1	0	0	1	9	DRAWING (Blink OFF Transp ON)	MARKER ON No Reposition		
1	0	1	0	10	DRAWING (Blink ON Transp OFF)	MARKER OFF Reposition		
1	0	1	1	11	DRAWING (Blink ON Transp ON)	MARKER ON Reposition		
1	1	0	0	12	TONAL CONTROL (Colour)	WAIT		
1	1	0	1	13	CONTROL STATUS	TONAL CONTROL (Grey)	WAIT (model)	
1	1	1	0	14		TONAL CONTROL (Reserved)	WAIT (flow control)	
1	1	1	1	15		TONAL CONTROL (Reserved)	WAIT (indef flow control)	

Figure 6. Table of Status Commands Accessed via the CONTROL (STATUS) OPCODE

There are basically two types of interaction functions, those associated with character coded devices such as the KEYBOARD or KEYPAD and those associated with graphical devices such as a LOCATOR (i.e., a tablet or a joystick device). In current Videotex applications only the character coded devices are of interest but room must be left in the coding scheme to allow for future expansion.

The KEYBOARD device permits a Videotex terminal to behave in an exactly compatible manner to the operation of a business or computer terminal. This means that each character which is typed is immediately transmitted back to the host computer.

The KEYPAD device is used to access information from a Videotex data base. A message is accumulated within a terminal and is composed of a string of Pushbutton presses. This string is transmitted back to the host computer upon the user pressing an activation button.

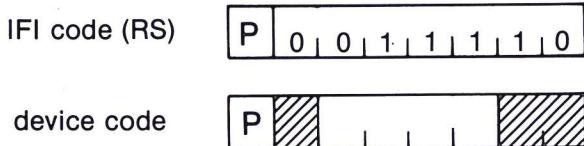
Control codes for function identification are not desirable in KEYBOARD interactions if one wishes to maintain compatibility with existing business terminals. In other words, the KEYBOARD function is the default condition for all text communication services. This means that all other functions must be identified by special control codes. These special control codes are similar to transmission control codes, but cannot be chosen from existing transmission control codes from the CO set if compatibility with existing business terminals and international standards is to be maintained.

The response coding scheme used by a Videotex terminal must allow for the distinction between these classes of interaction as well as leave room for additional future devices and nonstandard private devices. This means that a code identifying the device must be included with every change of device type.

In order to maintain compatibility with conventional business terminals, the response codes for the KEYBOARD form of the interaction code must be identical to the keystrokes entered by the user. Characters must be encoded according to ISO standards in the same manner as the GØ and CØ control sets. This means that if the character A, for example, is typed on a keyboard, then the code from code table location 1/4 is transmitted.

The interactive device of prime importance in present day Videotex systems is the Keypad with its PUSHBUTTON function. A Keypad may typically include pushbuttons for the digits Ø-9, function buttons for forward and backwards tracing in Videotex systems, buttons to change from "User Identification" to "Menu Select" or "Cursor" mode, function separator buttons, and buttons for "Go" or "Abort". Many of these buttons are for local operation. Ideally terminals should assemble complete keypad messages before transmission in order to minimize the loading at the central host computer.

A message is sent back to the host computer when the interactive device type changes. This message is of the form of an interaction function identifier character (IFI) followed by a device code.



The IFI code has been assigned the code table position of RS record separator from the CØ set as its function is to separate device information records.

The device code is an integer number, packed into a byte as illustrated. Several bits in this byte are reserved for future extension.

The default devices are:

- 1 - Keyboard; in which activation is upon every character and there is no local echo;
- 2 - Keypad; in which activation is record oriented and is transmitted upon an activation (GO) button. This activation button is coded as a CR character and is included in a keypad response message. Echo of the keypad keystrokes is done upon the display screen.

The codes transmitted back to the data base computer representing the buttons on the keypad are the GØ character codes associated with each button symbol. The symbols up arrow, left and right arrow and ditto make use of the codes '^<>' respectively.

The codes from the Keypad are echoed on the screen in the local echo area. This area is at the location of the MARKER (number Ø). The MARKER may be left on to indicate this spot or left off.

6. CONCLUDING REMARKS

Great care has been used in the design of the Picture Description Instructions to free the coding scheme from the constraints of particular hardware implementation techniques, so that more sophisticated terminals may be introduced in the future. Room to grow has been deliberately designed into the system with respect to the resolution, speed and sophistication of the terminal.

Another dimension of growth is the addition of new features not previously accommodated. Compatibility with installed inventories of television and Videotex equipment can be achieved if reasonable defaults are established for terminals which do not support these optional features. It is impossible to attempt to predict all future avenues by which a Videotex service can be extended, but it is necessary to provide means by which extensions can take place. For this reason, a spare opcode has been left in

the basic PDI coding scheme. In addition there exist spare Status Commands accessed via the CONTROL (STATUS) opcodes to allow more system status features to be developed.

- a) Line Width could provide an additional dimension to drawing commands but has not been defined for the basic PDI set because it would involve geometric calculations in the terminal which are beyond those required by conventional Videotex terminals. Trigonometric calculations are required in defining line thickness to ARCs, POLYGON outlines, etc. This is left to future terminals which would have more sophisticated microprocessors.
- b) Colour Lookup could provide the capability of defining specific colours such as flesh tones or shades of yellow to brown to extend a terminal's colour capability. Additional hardware is required for this capability which is not cost effective at this time.
- c) A feature which could reduce the load on the communications component and central data base computer of a Videotex system, is the ability for a terminal to receive a number of pages of information in a block transfer and to examine this "magazine" off-line from the central computer. A terminal would require secondary storage for pages composed of PDI commands. Current developments in bubble memory and other secondary storage media make this an attractive avenue of development for the next generation Videotex systems.
- d) An enhancement of the above feature would be the off-loading of even more tasks to increasingly more sophisticated terminals. The term "telesoftware" has been used to describe the dynamic loading of programs into terminals. This would have particular application to Computer Aided Learning and video games. Great care must be taken to ensure that any "telesoftware" programming system be compatible for all types of terminals. One possible method which could be suggested is a system of "Sequences" and sequence control be established. A sequence would consist of a group of PDI commands along with similarly encoded commands for invoking and delimiting sequences and commands to perform logical conditional operations. The implementation of a "Sequence" feature would require dynamically allocatable "telesoftware" storage space in the terminal and an increase in the complexity of the terminal PDI processing software beyond what current economics would allow. However, it is a powerful feature which could be implemented in future systems.
- e) A far more sophisticated feature for a home or office Videotex terminal is the capability of communicating directly from terminal to terminal. The Communications Research Centre has been involved in research in this area for some time and has identified techniques by which only a minimum amount of information needs to be communicated between the terminals. To this end, Picture Manipulation Instructions have been suggested as a portion of the CONTROL commands, so that pictures may be modified simultaneously at many terminals.

CRC DOCUMENT CONTROL DATA

1. ORIGINATOR: Department of Communications/Communications Research Centre

2. DOCUMENT NO: CRC Technical Note No. 699-E

3. DOCUMENT DATE: November 1979

4. DOCUMENT TITLE: Picture Description Instructions
PDI
For the Telidon Videotex System

5. AUTHOR(s): H.G. Bown, C.D. O'Brien, W. Sawchuk and J.R. Storey

6. KEYWORDS: (1) PDI
(2) Telidon
(3) Videotex

7. SUBJECT CATEGORY (FIELD & GROUP: COSATI)

14 Methods and Equipment

14 07 General Concepts

8. ABSTRACT:

The Telidon Videotex system is a method by which information can be accessed from central data bases by the general public. By the use of a domestic home television receiver augmented by a micro-computer controlled interface device a user can access pages of graphical and textual information over a public common carrier communications facility such as the telephone network, a cable-television line or the data may be even encoded into unused space in a broadcast television signal.

In order to transmit information to a Telidon terminal, at minimum bandwidth, and in a manner independent of the type of communications channel, a coding scheme was devised which encodes a picture into the geometric drawing elements which compose it. These "Picture Description Instructions" are an alpha-geometric coding model and are based on the primitives of POINT, LINE, ARC, RECTANGLE, POLYGON, point by point BIT encoded images and TEXT encoded as ASCII characters. This document provides a detailed specification of this code as well as a description of the principles which make it independent of communications channel and display hardware.

9. CITATION: _____



Government
of Canada Gouvernement
du Canada