



This electronic version (PDF) was scanned by the International Telecommunication Union (ITU) Library & Archives Service from an original paper document in the ITU Library & Archives collections.

La présente version électronique (PDF) a été numérisée par le Service de la bibliothèque et des archives de l'Union internationale des télécommunications (UIT) à partir d'un document papier original des collections de ce service.

Esta versión electrónica (PDF) ha sido escaneada por el Servicio de Biblioteca y Archivos de la Unión Internacional de Telecomunicaciones (UIT) a partir de un documento impreso original de las colecciones del Servicio de Biblioteca y Archivos de la UIT.

(ITU) للاتصالات الدولي الاتحاد في والمحفوظات المكتبة قسم أجراه الضوئي بالمسح تصوير نتاج (PDF) الإلكترونية النسخة هذه والمحفوظات المكتبة قسم في المتوفرة الوثائق ضمن أصلية ورقية وثيقة من نقلًا.

此电子版（PDF版本）由国际电信联盟（ITU）图书馆和档案室利用存于该处的纸质文件扫描提供。

Настоящий электронный вариант (PDF) был подготовлен в библиотечно-архивной службе Международного союза электросвязи путем сканирования исходного документа в бумажной форме из библиотечно-архивной службы МСЭ.



INTERNATIONAL TELECOMMUNICATION UNION

# CCITT

THE INTERNATIONAL  
TELEGRAPH AND TELEPHONE  
CONSULTATIVE COMMITTEE

**ANNEX TO RED BOOK**

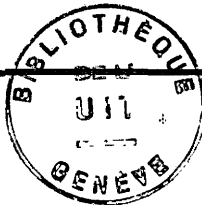
---

**FASCICLE VII.3 – ANNEX I**

## **DATA SYNTAX I FOR INTERNATIONAL INTERACTIVE VIDEOTEX SERVICE**

**RECOMMENDATION T.101, ANNEX B**

---



**VIII<sup>TH</sup> PLENARY ASSEMBLY**  
MALAGA-TORREMOLINOS, 8-19 OCTOBER 1984

Geneva 1985



INTERNATIONAL TELECOMMUNICATION UNION

# CCITT

THE INTERNATIONAL  
TELEGRAPH AND TELEPHONE  
CONSULTATIVE COMMITTEE

**ANNEX TO RED BOOK**

---

**FASCICLE VII.3 – ANNEX I**

## **DATA SYNTAX I FOR INTERNATIONAL INTERACTIVE VIDEOTEX SERVICE**

**RECOMMENDATION T.101, ANNEX B**

---



**VIII<sup>TH</sup> PLENARY ASSEMBLY**

MALAGA-TORREMOLINOS, 8-19 OCTOBER 1984

Geneva 1985



ANNEX B

(to Recommendation T.101)

DATA SYNTAX I

Note : This data syntax generally corresponds to the "CAPTAIN" presentation layer data syntax officially adopted by Japan.

Preface

This Annex describes the formats, rules and procedures for encoding of textual, pictorial and musical information for videotex applications. This specification conforms to the architecture defined in ISO's and CCITT's multi-layered reference model of open systems interconnection as part of a presentation level protocol.

Operation is accommodated in both a 7-bit and an 8-bit environment. For textual information, alphanumerics, Japanese characters and the Dynamically Redefinable Character Sets (DRCSs for both one byte and two bytes) are provided. For pictorial information, the photographic coding scheme, which is a distinctive feature of this data syntax, is provided. The mosaic coding and geometric coding are also available for pictorial information. There are two mosaic sets: One mosaic set is compatible with that of Data Syntax II. The geometric coding scheme conforms to Data Syntax III. Additional capabilities include colour look-up table, macros, flashing, unprotected fields, concealed characters, variable character size and scrolling. Musical note encoding and animated picture coding are optionally available.

## Contents

	<u>Page</u>
1. General	7
1.1 Scope	7
1.2 References	8
1.3 Definitions	8
1.4 Visual information display	11
1.4.1 Display structure	11
1.4.2 Logical frame structure	12
1.4.2.1 Block colouring frame	12
1.4.2.2 Dot colouring frame	12
1.4.3 Coordinate system	14
1.5 Musical information	14
2. Coding architecture	16
2.1 General	16
2.2 Mode switching technique	16
2.3 Coding structure in the character code mode	16
2.4 Coding structure in the transparent mode	16
2.5 Coding structure in the musical mode	21
3. Coding in the character code mode	22
3.1 C0 control set	22
3.2 C1 control set	25
3.2.1 Colour controls	26
3.2.2 Character size controls	28
3.2.3 Cursor controls	29
3.2.4 Flashing controls	29

	<u>Page</u>
3.2.5 Repeat controls	29
3.2.6 Lining controls	30
3.2.7 Conceal controls	30
3.2.8 Protect control	30
3.2.9 Photo macro command	31
3.2.9.1 General	31
3.2.9.2 P-DEF MACRO	31
3.2.9.3 P-DEFP MACRO	31
3.2.9.4 P-DEFT MACRO	31
3.2.9.5 P-MACRO END	32
3.3 Display control command set	33
3.3.1 General	33
3.3.2 Attribute control functions	37
3.3.2.1 P-DOMAIN	37
3.3.2.2 LOGICAL PEL	39
3.3.2.3 Colour look up table controls	39
3.3.2.3.1 SET LUT	39
3.3.2.3.2 P-BLINK	42
3.3.2.4 Raster colour controls	42
3.3.2.5 P-RESET	43
3.3.2.6 DISPLAY MODE	45
3.3.2.7 P-TEXT	46
3.3.2.8 P-WAIT	51
3.3.3 Multi-frame control function	51
3.3.3.1 General	51

	<u>Page</u>
3.3.3.2 General flow of multi-frame display operations	52
3.3.3.3 AREA	52
3.3.3.4 SET FRAME	55
3.3.3.5 ASSIGN FRAME	55
3.4 Character sets	57
3.4.1 Japanese-Kanji character set	57
3.4.2 Primary character set	57
3.4.3 Katakana character set	57
3.5 Mosaic sets	57
3.5.1 Mosaic 1 set	57
3.5.2 Mosaic 2 set	57
3.6 Dynamically Redefinable Character Sets (DRCSs)	65
3.6.1 DRCS 1	65
3.6.2 DRCS 2	65
3.7 Macro set	65
3.8 Picture Description Instruction (PDI) set	65
3.9 Moving Instruction (MVI) set	68
3.9.1 MVI START	70
3.9.2 MVI END	70
3.9.3 REPEAT START	70
3.9.4 REPEAT END	70
3.9.5 MOVE	71
4. Coding in the transparent mode	72
4.1 Photographic data unit	72



	<u>Page</u>
4.2 LINE DOT PATTERN	73
4.3 LINE DOT PATTERN COMPRESSED	73
4.4 FIELD DOT PATTERN	76
4.5 COLOURING BLOCK	80
4.6 COLOURING BLOCK COMPRESSED	83
4.7 FIELD COLOURING BLOCK	83
4.8 FIELD COLOURING BLOCK COMPRESSED	88
4.9 FREE FORMAT COLOURING BLOCK	90
4.10 PHOTO DRCS 1 and PHOTO DRCS 2	90
5. Coding in the musical mode	95
5.1 General	95
5.2 Musical tone set	96
5.3 Musical control set	96
5.3.1 Start and end of music codes	96
5.3.1.1 SMC	96
5.3.1.2 EMC	96
5.3.2 Start and end of a part	100
5.3.2.1 SMP	100
5.3.2.2 SRP	101
5.3.2.3 EPT	101
5.3.3 Repeat and jump	101
5.3.3.1 LBL	102
5.3.3.2 JMP	102
5.3.3.3 RPT	102
5.3.3.4 BRA	102

	<u>Page</u>
5.3.4 SLV	102
5.3.5 CTM	103
5.3.6 LRT	103
5.3.7 Numeric parameter	104
6. Default conditions	105
Appendix 1 Service Reference Model	107
Appendix 2 Operation of CS and NSR	109

## 1. General

### 1.1 Scope

This standard describes the Data Syntax I, Annex B of draft Recommendation T.101, which provides means for exchanging textual, pictorial and audio information in the international videotex service interworking.

## 1.2 References

This standard is intended to be as closely compatible as possible with the following recommendations and standards:

CCITT* Recommendation T.50	International alphabet No.5
CCITT Recommendation F.300	Videotex service
CCITT Recommendation S.100	International information exchange for interactive videotex
ISO** Standard 2022 - 1982.	ISO 7 bit 8 bit coded character sets-code extension techniques.
ISO Standard 6937 - 1982.	Information processing - coded character sets for text communication

\* International Telegraph and Telephone Consultive Committee.

\*\* International Organization for Standardization.

## 1.3 Definitions

The following definitions apply in this standard:

**Attribute** means a settable parameter to be applied to subsequent textual, pictorial or audio information.

**Bit combination** is an ordered set of bits (binary digits) that represents a character or a control function.

**C-set** stands for control set. There are two control sets, C0 and C1, each of which comprises 32 character positions arranged in 2 columns by 16 rows.

**Character block** means the rectangular area within which a normal size character is displayed.

**Character code mode** is the mode in which the coding structure is based on the code extension technique of ISO 2022.

**Character code mode command** means a character or a command such as one in the Display Control Commands which is encoded in the character code mode.

**Character field** means the rectangular area within which a charac-

ter is displayed in the currently defined character size.

Code extension means techniques for expanding the absolute character address space of a byte-oriented code into a larger virtual address space.

Code table means the set of unambiguous rules that defines the mapping between received bit combinations and presentation level characters.

Defined display area means the addressable area of the physical display screen onto which the unit screen or a portion of the unit screen is mapped. Header area is not included in this area.

Designate means to identify a given set from the repertory of G-sets as a G0, G1, G2, or G3 set.

Display Control command set is a G-set. A control command is composed of an opcode following by zero or more operands and defines an attribute control function or a multi-frame structure.

Dynamically Redefinable Character Set (DRCS) is a G-set containing definable characters whose patterns can be downloaded from the host.

Escape sequence means a string of two or more bit combinations beginning with the ESC character. Formats and rules regarding the use of the escape sequences are specified in ISO 2022.

Frame is the minimum unit of the display structure, on which complete information, but not necessarily whole information, can be displayed. Entire information on the screen may be composed of a few frames. Typical implementation of a frame requires a few memory planes.

G-set refers to one of the four sets, G0, G1, G2 and G3, each of which comprises 94 or 96 character positions arranged in 6 columns by 16 rows.

Header Area is used for the display of system messages and key pad input monitor.

Japanese-Kanji character set is a 2 bytes G-set. The total or 3657 characters including Japanese-Kanji characters, Japanese phonetic signs (Katakana and Hiragana), Roman characters, numerics and additional characters are defined.

Katakana character set is a G-set which defines 63 Japanese phonetic signs.

Layer is terminology adopted by ISO to describe each individual module of the reference model for open systems interconnection (OSI) (the terms "level" and "layer" are used interchangeably).

Locking shift means an invocation of a code set into the in-use table that remains in effect until another code set is invoked in its place.

Mosaic is a rectangular matrix of pre-defined elements that can be used to construct block-style graphic images.

Move Instruction (MVI) is composed of an opcode followed by zero or more operands and constitutes an executable frame moving or control command.

Musical control set is a C-set. The control codes are used for controlling musical sound generation.

Musical mode is the mode in which musical information is carried. Musical tones are encoded with the character coding techniques.

Musical Tone set is a two byte G-set which defines pitch and duration of a musical sound.

Photographic Data Unit (PDU) is composed of an opcode with a length indicator (LI), followed by zero, one or more operands, each of which consists of one or more octets of bit sequences. This use of all possible octet patterns are allowed in the operands bit sequences, which results in efficient expression or arbitrary data. The opcode consists of a single octet which indicates the meaning of photographic data contained in the PDU. The LI consists of one or more octets. The value of an LI is a binary number that represents the total length of operands following the LI field in octet. One or more octets of parameters are located at the leading part of an operand field. Parameters include the drawing point coordinates where the photographic data should be displayed, and/or the packing format which indicates the way in which photographic data are arranged. Photographic data expressed on a dot-by-dot basis are contained in the remaining part of the operand field.

Picture description instruction (PDI) is composed of an opcode followed by zero or more operands and constitutes an executable picture drawing or control command.

Plane (or memory plane) is a component part of a frame, accommodating an information component such as pattern information, foreground colour information, etc..

Presentation level (or Layer) is the sixth of seven protocol levels defined by ISO's reference model of open systems interconnection. The presentation level is primarily responsible for the encoding of text, graphic and display control information.

Primary character set is a G-set which defines 52 Roman characters, 10 numerics and 32 marks.

Protocol is a set of formats, rules and procedures governing the exchange of information between peer processes at the same level.

Service Reference Model (SRM) defines the recommended features that should be implemented by an ordinary terminal or decoder.

Single shift is an invocation of a code set into the in-use table that affects only the interpretation of the next bit combination received. Interpretation then automatically reverts to the previous contents of the table. (This is also referred to as non-locking shift).

Terminal equipment is equipment that can exchange coded bit combinations by means of telecommunication or by physical interchange of storage media.

Transparent mode is the mode in which all the presentation level bits are used for pictorial, audio or telesoftware data. This mode provides an efficient means of transmitting the relatively large amount of data.

Transparent mode command means a command encoded in the transparent mode.

Unit screen means the logical display address space within which all drawing operations are executed and alphanumeric characters are deposited. The dimensions of the unit screen are 0 to 1 in the horizontal (X), vertical (Y) and depth (Z) dimensions. (The last is only defined in three-dimensional mode.)

Visible display area means the entire physical display screen visible to the user.

#### 1.4 Visual information display

##### 1.4.1 Display structure

This standard allows the multi-frame structure which provides superimposition of several frames. Fig.1.1 illustrates

an example of multi-frame structure.

Each frame can be of different constructions: A dot colouring frame has colour information on a dot-by-dot basis, while a block colouring frame has one on a block-by-block basis.

The full screen background layer has the lowest display priority: The Raster colour and the Header Raster colour are displayed only where an upper frame is "transparent" and outside of the defined display area.

Multi-frame manipulation is achieved through some display control commands. Details are given in section 3.3.3.

#### 1.4.2 Logical frame structure

##### 1.4.2.1 Block colouring frame

A block colouring frame is composed of a pattern plane, a foreground colour (FG) plane, a background colour (BG) plane and display attribute (DA) plane as illustrated in Fig.1.1.

Colour information in the FG plane and the BG plane is given on a colouring block basis. The colouring block is a rectangular area, normally larger than the pattern plane's pixel, and is a minimum unit of colour definition.

The pattern plane is a one-bit-deep memory plane. Where the bit is '1', the foreground colour is displayed. Where the bit is '0', the background colour is displayed.

A display attribute plane has flags for flashing, conceal and protect attributes on a colouring block basis.\*

The block colouring frame structure economizes both data transmission time and the amount of a raster memory at the expense of colour resolution. Text and pictures are normally drawn on this type of frame.

##### 1.4.2.2 Dot colouring frame

A dot colouring frame can store colour information on a pixel basis. Geometrically coded pictures are best reproduced on a dot colouring frame. (See Fig.1.1.)

---

\* Flashing: Patterns are displayed alternately in the foreground colour and the background colour.

Conceal: Patterns are made invisible until the conceal attribute is cancelled.

Protect: A character block or a colouring block is protected against alternation by a user.



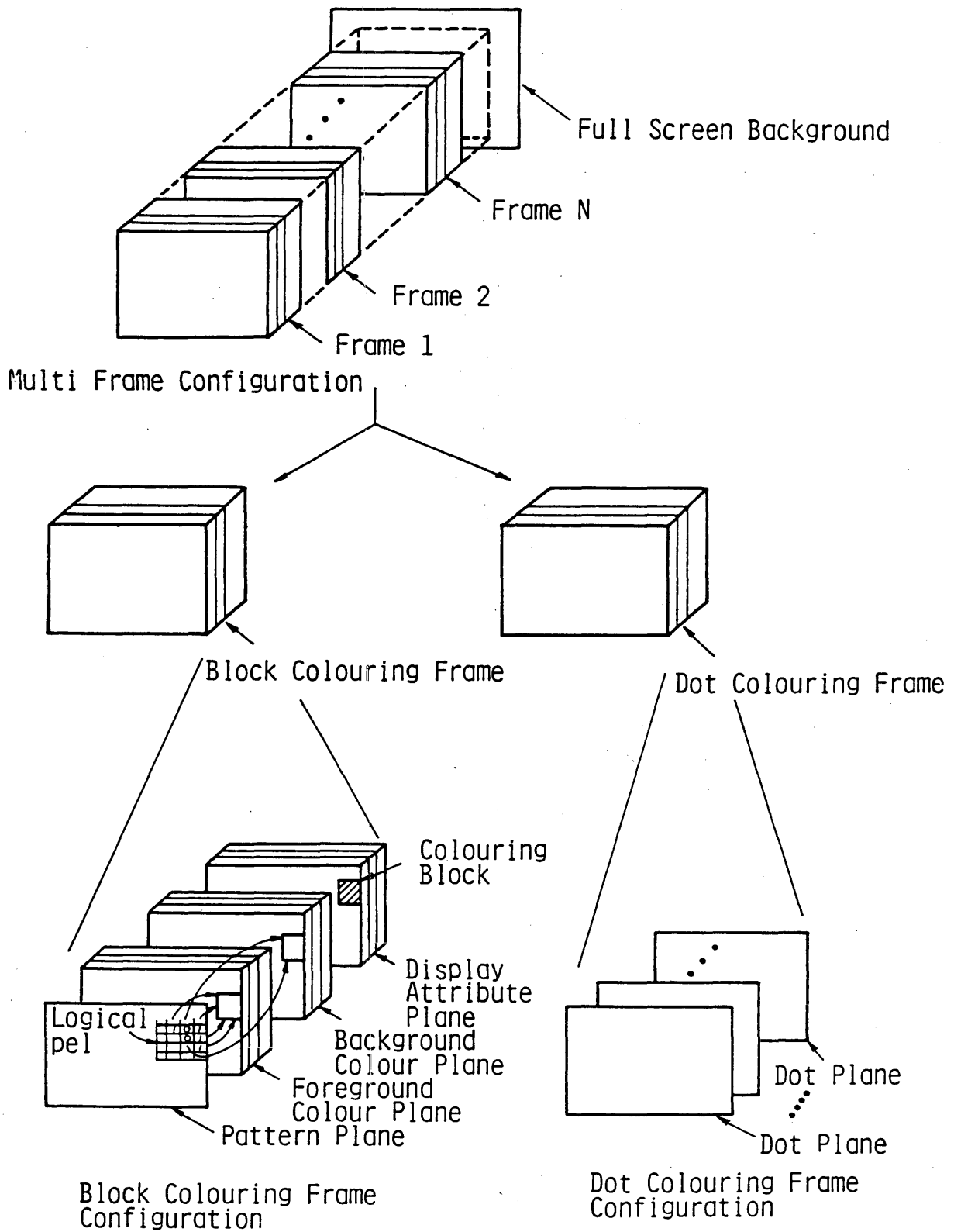


Fig.1.1 Multi-Frame Structure Concept

#### 1.4.3 Coordinate system

The coordinate system utilized in both photographic and geometric picture data is based on the unit screen concept. A logical pel or a geometric primitive is always located within the unit screen, whose horizontal (X) and vertical (Y) coordinates range from 0 to 1. The origin (0, 0) of the unit screen is mapped to the lower left corner of the physical display screen.

The physical display screen should cover at least the rectangle region  $0 \leq x \leq 0.969$ ,  $0 \leq y \leq 0.797$ . Normal information is displayed within the region  $0 \leq y \leq 0.75$ , while the remainder of the screen ( $0.75 \leq y \leq 0.797$ : Header) is used for the display of system messages and key pad input monitor.

The full screen background layer is divided into two regions; Raster ( $y \leq 0.75$ ) and Header raster ( $0.75 \leq y$ ).

The concept of the unit screen and full screen background layer is shown in Fig.1.2 and 1.3.

#### 1.5 Musical information

Musical information is encoded into pseudo character codes and is downloaded to a terminal. Then a multi-channel tone generator installed in the terminal can play music.

Mode switching technology is employed to present musical information. (See Section 2.2.)

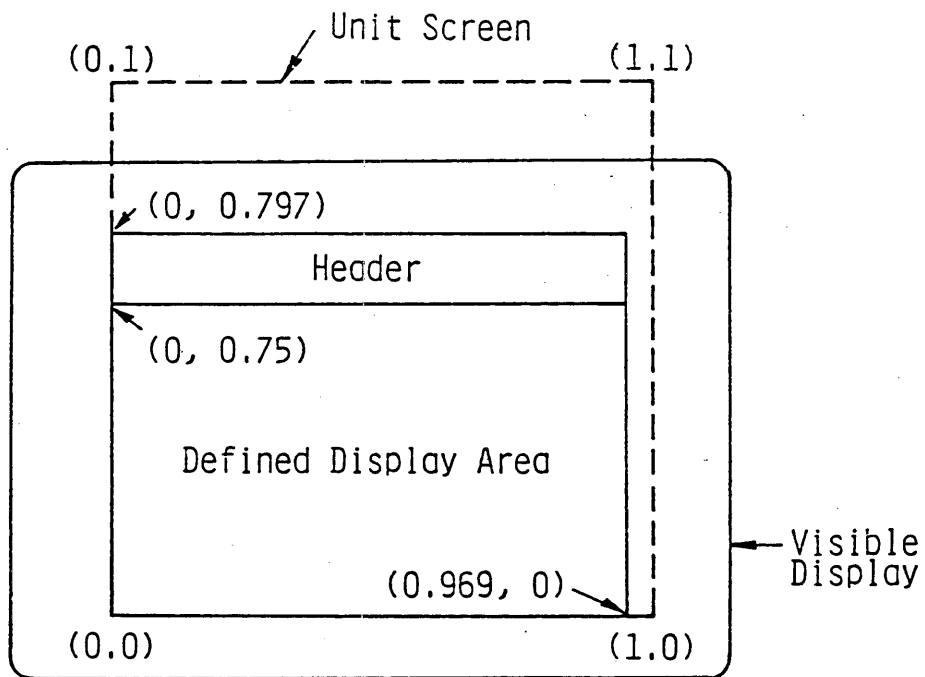


Fig.1.2 Unit Screen Concept

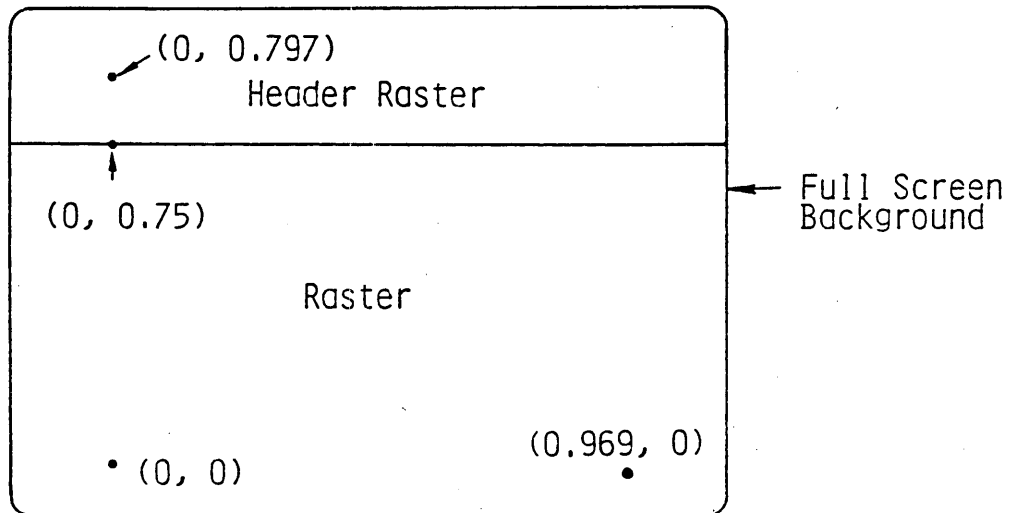


Fig.1.3 Full Screen Background Structure

## 2. Coding architecture

### 2.1 General

Textual information is character coded and geometric primitives, musical notes and display control commands are encoded in the same scheme as character codes. Both 7 bit and 8 bit environments are supported.

On the other hand, photographic pictorial data is encoded in a transparent coding scheme in order to store and transmit the relatively large amounts of data efficiently.

### 2.2 Mode switching technique

This standard specifies three separate operation modes: The character code mode, the transparent mode and the musical mode. The character code mode is the mode in which the coding structure is based on the code extension technique of ISO 2022.

In the transparent mode, transparent data can be conveyed.

In the musical mode, musical notes are encoded and transmitted with the character coding technique, however a C-set and a G-set are different from those in the character code mode.

The default mode is the character code mode. The transparent mode is invoked by the start transparent mode initiator (STM) which itself is based on a character coding method. The character code mode is regained by the length indicator and/or through the end transparent mode terminator (ETM).

The musical mode is invoked by the start musical mode (SMM) initiator and terminated by the end musical mode (EMM) terminator. Exact code assignment for STM, ETM, SMM and EMM is subject to the ISO standardization.

Fig.2.1 illustrates the mode switching concept.

### 2.3 Coding structure in the character code mode

The coding structure in the character code mode is based on the code extension principles of ISO 2022 for both the 7 bit and 8 bit environments. Fig.2.2 shows the code extension method of Data Syntax I. Escape sequences for designation of C-sets and G-sets are shown in Table 2.1.

### 2.4 Coding structure in the transparent mode

In the transparent mode, transparent data are separated into Photographic Data Units (PDUs). The structure of a PDU is shown in Fig.2.3. The first octet (8 bits) of a PDU is an opcode which defines the meaning of the PDU. The second octet is

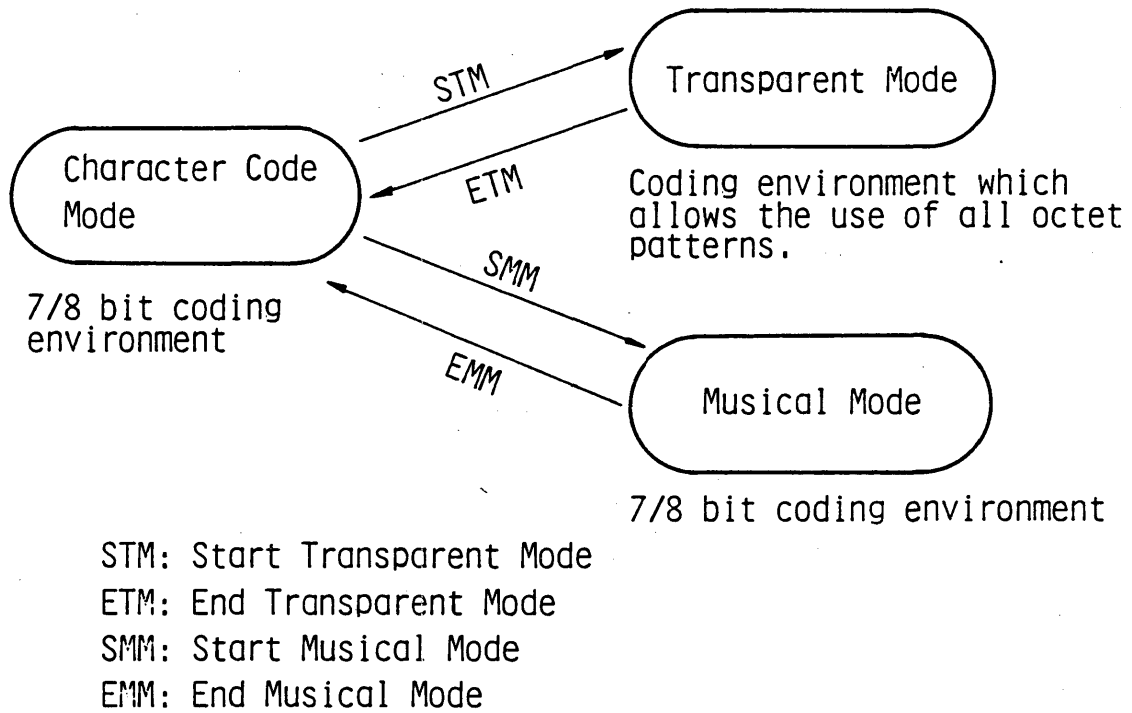


Fig.2.1 Mode Switching Concept

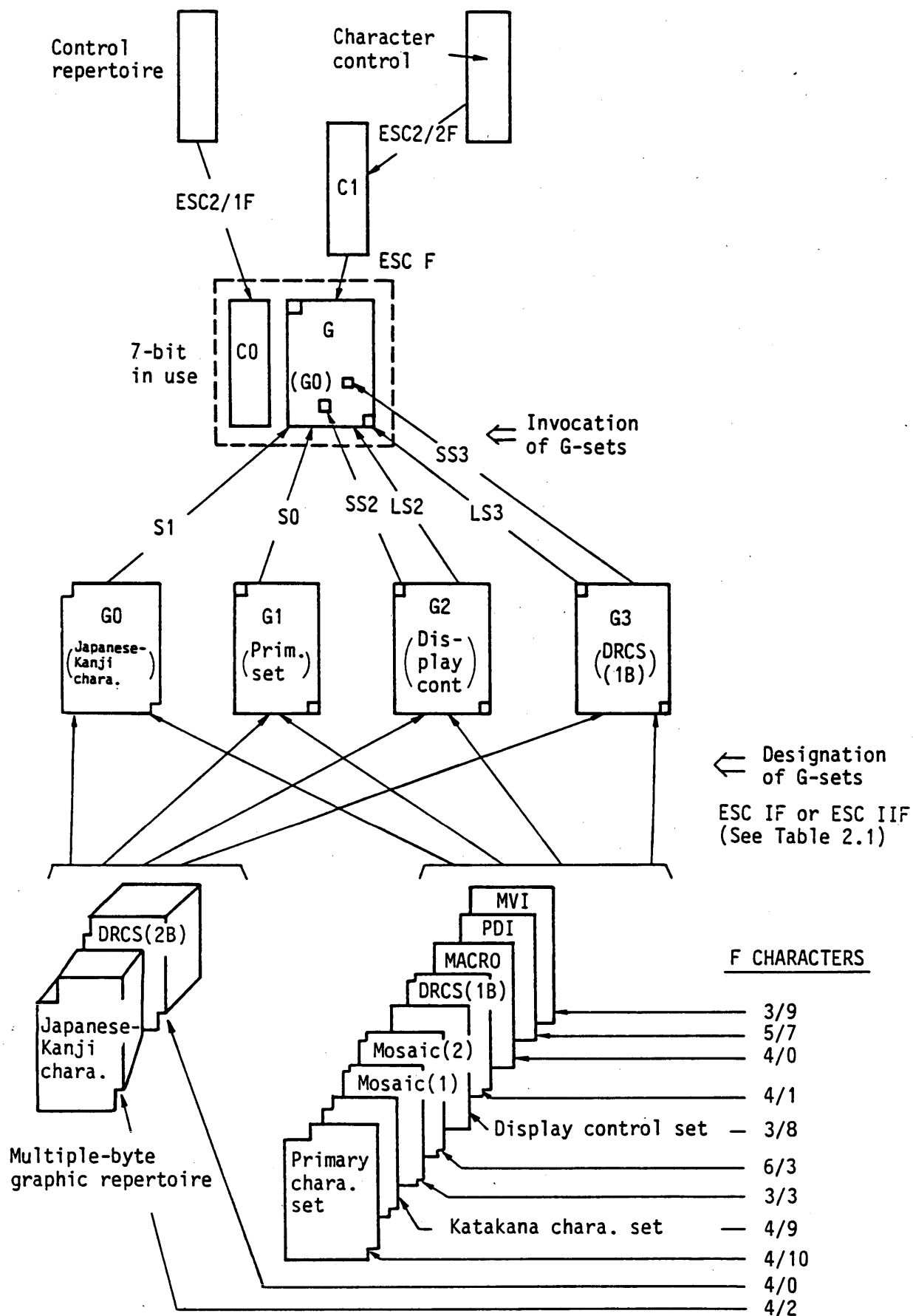


Fig.2.2(1) Code Extension in the Character Code Mode (in 7-bit environment)

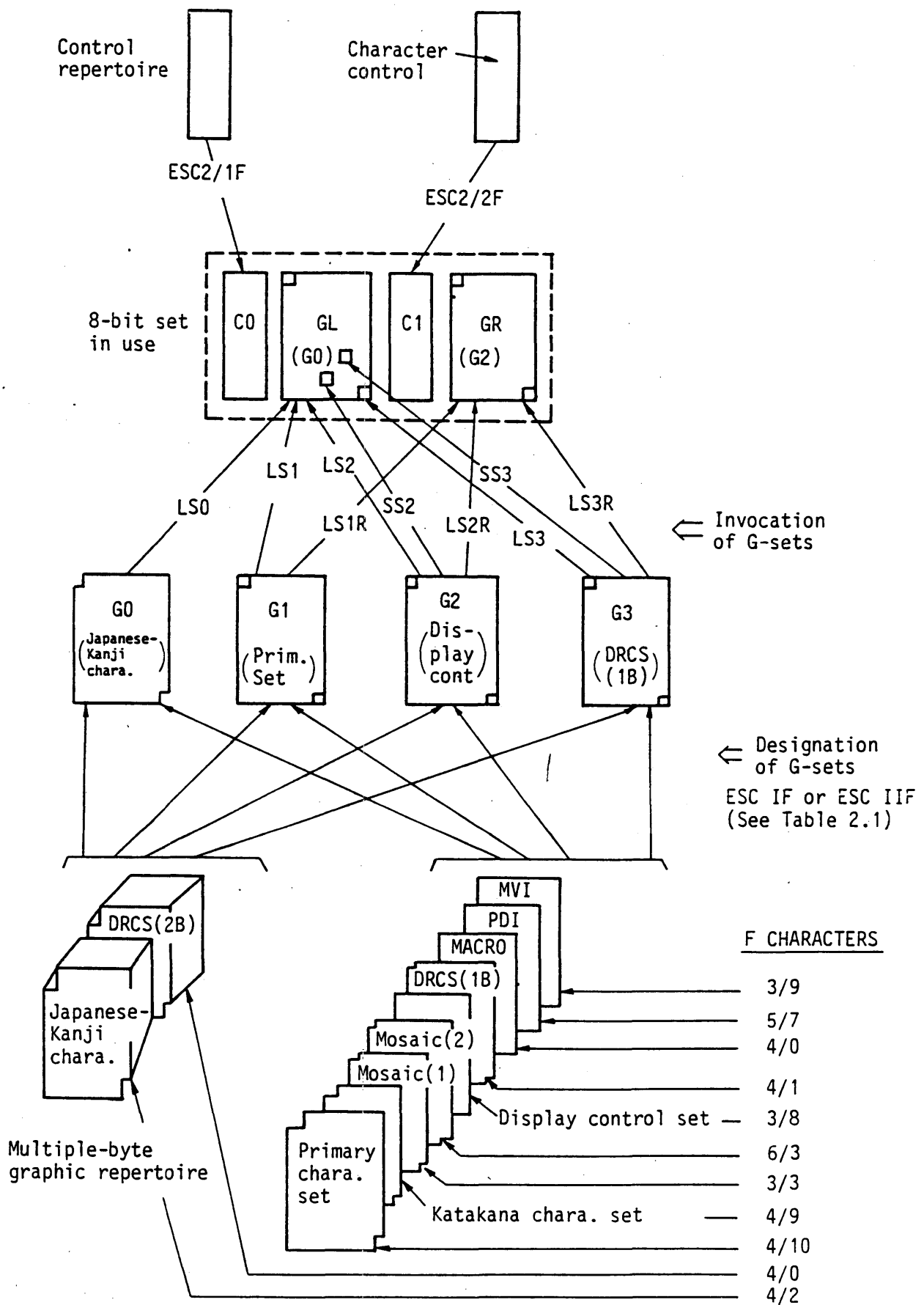
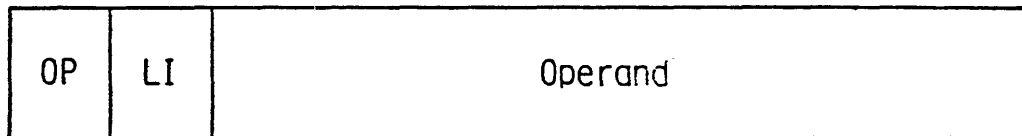


Fig.2.2(2) Code Extension in the Character Code Mode (in 8-bit environment)

Table 2.1 Escape Sequences for Designation of C-sets and G-sets

Escape Sequence	Set to be Designated
Control sets:	
ESC 2/1 F3	C0 set
ESC 2/2 F4	C1 set
94-Character sets:	
ESC I 4/10	Primary Character set
ESC I 4/9	Katakana Character set
ESC I 3/3	Mosaic 1 set
ESC I 3/2	Mosaic 2 set
ESC I 2/0 4/1	DRCS 1
Where I is 2/8, 2/9, 2/10, 2/11 for G0, G1, G2, G3 respectively.	
96-Character sets:	
ESC I 3/8	Display control set
ESC I 5/7	PDI set
ESC I 3/9	MVI set
ESC I 2/0 4/0	MACRO set
Where I is 2/13, 2/14, 2/15, for G1, G2, G3, respectively.	
Multiple-byte sets:	
ESC 2/4 I 4/2	Kanji character set
ESC 2/4 I 2/0 4/0	DRCS 2
( Where I is 2/9, 2/10, 2/11 for G1, G2, G3, respectively and I is not involved for G0. )	





OP: Opcode

LI: Length Indicator

Fig.2.3 Photographic Data Unit Format

a length indicator which indicates the number of octets in a PDU. If the decimal expression of the second octet is 255, then the next two octets (the third and the fourth) indicate the real length. After the length indicator, the necessary bit data sequence follows.

Presentation level data flow is illustrated in Fig.2.4.

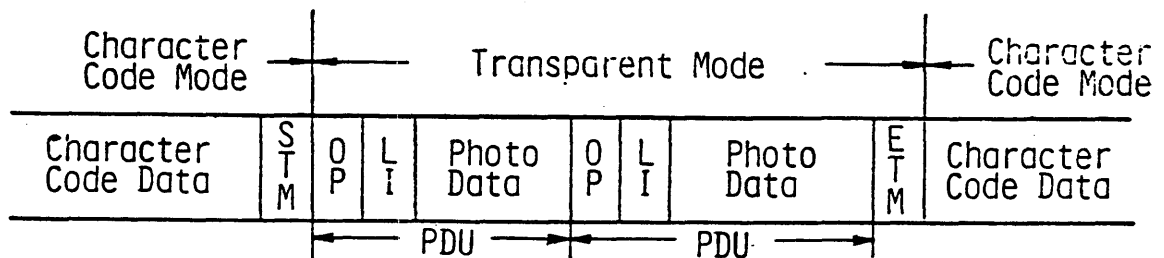


Fig.2.4 Presentation Level Data Flow (Transparent Mode)

## 2.5 Coding structure in the musical mode

Commands and parameters are character coded in the musical mode. However, a C-set and a G-set different from the character code mode are set upon invocation of the mode.

Presentation level data flow is illustrated in Fig.2.5.

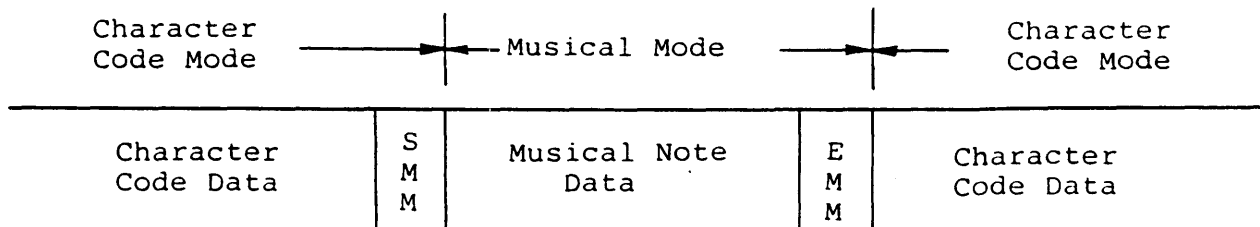


Fig.2.5 Presentation Level Data Flow (Musical Mode)

### 3. Coding in the character code mode

#### 3.1 C0 Control Set

The C0 control set is shown in Table 3.1. The functions are as follow:

Table 3.1 C0 Control Set

				b <sub>7</sub>	0	0
				b <sub>6</sub>	0	0
				b <sub>5</sub>	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		0	1
0	0	0	0	0	NUL	
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	BEL	
1	0	0	0	8	APB	CAN
1	0	0	1	9	APF	SS2
1	0	1	0	10	APD	
1	0	1	1	11	APU	ESC
1	1	0	0	12	CS	APS
1	1	0	1	13	APR	SS3
1	1	1	0	14	SO	APH
1	1	1	1	15	SI	NSR

- NUL (0/0) NULL character has no effect on the presentation level. It may be used for timing adjustment.
- BEL (0/7) BELL character is used to momentarily ring a bell for attracting the user's attention.
- CAN (1/8) CANCEL is used to terminate processing of all currently executing macros. Execution is resumed at the next presentation level character following the terminated macro call. The effect of the CAN is immediate.
- APB (0/8) ACTIVE POSITION BACKWARD is used to position the cursor a distance equal to the inter-character spacing lying parallel to the character path in the direction opposite to the character path. If such a movement would cause the edge of the defined display area to be crossed, then the cursor is positioned at the opposite edge of the active drawing area and an automatic APU is executed.
- APF (0/9) ACTIVE POSITION FORWARD is used to position the cursor a distance equal to the inter-character spacing lying parallel to the character path in the direction of the character path. If such a movement would cause the edge of the defined display area to be crossed, then the cursor is positioned at the opposite edge of the defined display area and an automatic APD is executed.
- APD (0/10) ACTIVE POSITION DOWN is used to position the cursor a distance equal to the inter-row spacing lying perpendicular to the character path in a direction perpendicular to the character path (-90 deg.). If such a movement would cause the edge of the defined display area to be crossed, the cursor is positioned at the opposite edge of the defined display area.
- APU (0/11) ACTIVE POSITION UP is used to position the cursor a distance equal to the inter-row spacing lying perpendicular to the character path in a direction perpendicular to the character path (90 deg.). If such a movement would cause the edge of the defined display area to be crossed, then the cursor is positioned at the opposite edge of the defined display area.
- APR (0/13) ACTIVE POSITION RETURN is used to position the cursor to the first character position within the defined display area along the character path and an automatic APD is executed.
- APH (1/14) ACTIVE POSITION HOME is used to position the cursor to the upper left character position in the defined display area.
- APS (1/12) ACTIVE POSITION SET is used to set the cursor position which is specified by the two bytes parameter immediately following an APS. The first byte represents the row address and the second byte does the column address. The address is obtained by taking the binary values comprising bits b6 through b1 with b6 being the MSB. (Each

byte is from columns 4 through 7.)

Exact APS operations are as follow: First, an automatic APH is executed. Then, the cursor is positioned N times inter-row spacing below and M times inter-character spacing right of the home position, where N is the number specified by the first byte, and M is the number specified by the second byte.

If any of the two bytes is out of columns 4 through 7, the APS character and the following two bytes are taken as NULs.

- NSR (1/15) NSR serves two functions. First, NSR resets non-selectively all the display attributes to their default states. (See Section 6.) Then the NSR character sets the cursor positions by the two bytes parameter immediately following NSR. The cursor positioning is the same as APS, except that APS sets the cursor according to the current inter-character spacing and inter-row spacing, while NSR sets the cursor according to their default values. Macro definitions and DRCS definitions are not cancelled by NSR.
- CS (0/12) CS resets all the display attributes to their default states and activates an automatic APH. Then, all macro definitions and DRCS definitions are cancelled. CS clears the entire screen.
- ESC (1/11) ESC introduces a code extension sequence.
- LS0 (SI) (0/15) Locking shift 0.
- LS1 (SO) (0/14) Locking shift 1.
- SS2 (1/9) Single shift 2.
- SS3 (1/13) Single shift 3.

3.2 C1 control set

The C1 control set is used to allow control over textual display attributes and also to define macros. Table 3.2 shows the C1 control set table.

Table 3.2 C1 Control Set

				b <sub>8</sub>	1	1
				b <sub>7</sub>	0	0
				b <sub>6</sub>	0	0
				b <sub>5</sub>	0	1
					8	9
				b <sub>7</sub>	1	1
				b <sub>6</sub>	0	0
				b <sub>5</sub>	0	1
					4	5
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		4/8	5/9
0	0	0	0	0	BKF	COL*
0	0	0	1	1	RDF	FLC*
0	0	1	0	2	GRF	CDC*
0	0	1	1	3	YLF	
0	1	0	0	4	BLF	
0	1	0	1	5	MGF	**P-MACRO*
0	1	1	0	6	CNF	
0	1	1	1	7	WHF	
1	0	0	0	8	SSZ	RPC*
1	0	0	1	9	MSZ	SPL
1	0	1	0	10	NSZ	STL
1	0	1	1	11	SZX*	
1	1	0	0	12		
1	1	0	1	13		
1	1	1	0	14	CON	UNP
1	1	1	1	15	COF	PRT

\* This code has a parameter  
\*\* P-: Photographic

### 3.2.1 Colour controls

Each character on a screen has the foreground colour (pixels' colour where the value of the pattern plane is 1) and the background colour (pixels' colour where the value of the pattern plane is 0). Both the foreground colour and the background colour are specified by means of a colour look up table (LUT) entry address, however a code name is derived from the default colour of the LUT.

The default colours include six colours (red, green, yellow, blue, magenta, cyan) and black and white, and their reduced intensity versions. (See Table 3.5).

A full intensity foreground colour is specified by one of the following C1 control codes.

BKF (Black foreground) Invokes black or 0th colour of the LUT as a foreground colour.  
RDF (Red foreground) Invokes red or 1st colour of the LUT as a foreground colour.  
GRF (Green foreground) Invokes green or 2nd colour of the LUT as a foreground colour.  
YLF (Yellow foreground) Invokes yellow or 3rd colour of the LUT as a foreground colour.  
BLF (Blue foreground) Invokes blue or 4th colour of the LUT as a foreground colour.  
MGF (Magenta foreground) Invokes magenta or 5th colour of the LUT as a foreground colour.  
CNF (Cyan foreground) Invokes cyan or 6th colour of the LUT as a foreground colour.  
WHF (White foreground) Invokes white or 7th colour of the LUT as a foreground colour.

A full intensity background colour, a reduced intensity foreground colour or a reduced intensity background colour is specified by one byte parameter following the COL C1 control code.

COL 4/8 - TRF (Transparent foreground) Invokes transparent or 8th colour of the LUT as a foreground colour.  
COL 4/9 - RDFR (Red foreground reduced) Invokes reduced intensity red or 9th colour of the LUT as a foreground colour.  
COL 4/10 - GRFR (Green foreground reduced) Invokes reduced intensity green or 10th colour of the LUT as a foreground colour.  
COL 4/11 - YLFR (Yellow foreground reduced) Invokes reduced intensity yellow or 11th colour of the LUT as a foreground colour.  
COL 4/12 - BLFR (Blue foreground reduced) Invokes reduced intensity blue or 12th colour of the LUT as a foreground colour.  
COL 4/13 - MGFR (Magenta foreground reduced) Invokes reduced

- intensity magenta or 13th colour of the LUT as a foreground colour.
- COL 4/14 - CNFR (Cyan foreground reduced) Invokes reduced intensity cyan or 14th colour of the LUT as a foreground colour.
  - COL 4/15 - WHFR (White foreground reduced) Invokes reduced intensity white or 15th colour of the LUT as a foreground colour.
  - COL 5/0 - BKB (Black background) Invokes black or 0th colour of the LUT as a background colour.
  - COL 5/1 - RDB (Red background) Invokes red or 1st colour of the LUT as a background colour.
  - COL 5/2 - GRB (Green background) Invokes green or 2nd colour of the LUT as a background colour.
  - COL 5/3 - YLB (Yellow background) Invokes yellow or 3rd colour of the LUT as a background colour.
  - COL 5/4 - BLB (Blue background) Invokes blue or 4th colour of the LUT as a background colour.
  - COL 5/5 - MGB (Magenta background) Invokes magenta or 5th colour of the LUT as a background colour.
  - COL 5/6 - CNB (Cyan background) Invokes cyan or 6th colour of the LUT as a background colour.
  - COL 5/7 - WHB (White background) Invokes white or 7th colour of the LUT as a background colour.
  - COL 5/8 - TRB (Transparent background) Invokes transparent or 8th colour of the LUT as a background colour.
  - COL 5/9 - RDBR (Red background reduced) Invokes reduced intensity red or 9th colour of the LUT as a background colour.
  - COL 5/10 - GRBR (Green background reduced) Invokes reduced intensity green or 10th colour of the LUT as a background colour.
  - COL 5/11 - YLBR (Yellow background reduced) Invokes reduced intensity yellow or 11th colour of the LUT as a background colour.
  - COL 5/12 - BLBR (Blue background reduced) Invokes reduced intensity blue or 12th colour of the LUT as a background colour.
  - COL 5/13 - MGBR (Magenta background reduced) Invokes reduced intensity magenta or 13th colour of the LUT as a background colour.
  - COL 5/14 - CNBR (Cyan background reduced) Invokes reduced intensity cyan or 14th colour of the LUT as a background colour.
  - COL 5/15 - WHBR (White background reduced) Invokes reduced intensity white or 15th colour of the LUT as a background colour.

A foreground colour and a background colour specified remain unchanged until a new colour is specified. Execution of NSR or CS also resets colour specification.

The default foreground colour is white and the default background colour is transparent.

### 3.2.2 Character size controls

The following codes define a character size to be used. The character size, once defined, remains unchanged until it is altered by following C1 control set, a display control command (P\*-TEXT), NSR or CS.

- NSZ (Normal size) The character width and height are the same as that of the character block specified by the P-TEXT command described later.
- SSZ (Small size) The character width and height become half that of NSZ.
- MSZ (Medium size) The character width becomes half that of NSZ. The character height is the same as that of NSZ.
- SZX (Size control) SZX is the C1 control code which specifies a character size DBH, DBW and DBS by one byte parameter following the SZX code.
- SZX 4/1 - DBH (Double height): The character height becomes double that of NSZ. The character width is the same as that of NSZ.
- SZX 4/4 - DBW (Double width): The character width becomes double that of NSZ. The character height is the same as that of NSZ.
- SZX 4/5 - DBS (Double size): The character width and height become double that of NSZ.

The default dimensions of the character size are 'normal'.

\* P stands for 'photo'.



### 3.2.3 Cursor controls

CON (Cursor ON) The cursor display is made visible.

COF (Cursor OFF) The cursor display is turned off.

The default state is cursor off.

### 3.2.4 Flashing controls

Flashing is a process where a foreground colour is alternately turned into a background colour. The flashing attribute can be placed on the character size basis or on a colouring block basis. The default state is steady.

The FLC (Flashing control) code, which is one of the C1 control codes, specifies flashing attributes by one byte parameter following the FLC code.

FLC 4/0 - Normal flash: Applies the normal (50%) flash attributes.

FLC 4/7 - Inverted flash: Applies the inverted flash attributes.

FLC 4/4 - Three phase flash 1: Applies the 1st phase of three phase flashing.

FLC 4/2 - Three phase flash 2: Applied the 2nd phase of three phase flashing.

FLC 4/1 - Three phase flash 3: Applies the 3rd phase of three phase flashing.

FLC 4/3 - Three phase inverted flash 1: Applies the inverted 1st phase of three phase flashing.

FLC 4/5 - Three phase inverted flash 2: Applies the inverted 2nd phase of three phase flashing.

FLC 4/6 - Three phase inverted flash 3: Applies the inverted 3rd phase of three phase flashing.

FLC 4/15 - Steady: Cancels the application of any flash attributes.

The sequence FLC 4/0, A, B, C, D, FLC 4/15 causes characters "ABCD" to be normal-flashed.

### 3.2.5 Repeat controls

The RPC (Repeat control) code causes the following transmitted G set character, if the following character is a non-spacing character, both the nonspacing character and the next character, to be displayed a number of times specified by the following byte. The byte must be from columns 4 through 7. The repeat count is given by the binary number comprising bits b6 through b1 with b6 being the MSB. If the byte following the RPC code is not from columns 4 through 7, the RPC code is not executed. RPC 4/0 has a special meaning that repeat to end of line.

The sequence A, RPC 4/3, B, C causes "A BBB C" to be dis-

played.

### 3.2.6 Lining controls

The following codes control the application of the lining attribute:

STL (Start lining) Applies the lining attribute: Characters are displayed with an underline whereas mosaic patterns are displayed in separated font.

SPL (Stop lining) Resets the lining attribute.

### 3.2.7 Conceal controls

The CDC (Conceal display control) codes control the conceal display attribute. When the conceal display attribute is applied, characters and DRCS are made invisible. The character position takes ordinary advance. Concealed characters are made visible by the specified user action. The CDC code, which is one of the Cl control codes, specifies conceal display attributes by one byte parameter following the CDC code.

CDC 4/0 - CDY (Conceal display) Applies the conceal display attribute to the following characters.

CDC 4/15 - SCD (Stop conceal display) Resets the conceal display attribute.

The sequence A, B, C, CDC 4/0, D, E, F, ... , CDC 4/15 causes a terminal to display "ABC" first and "DEF ..." is made visible by user action.

### 3.2.8 Protect controls

Protected character blocks cannot be altered, manipulated or erased by user action. The entire screen is protected by default.

UNP (Unprotect) Following character are made unprotected.

PRT (Protect) Following characters are made protected.

Ex.

ABC (UNP) DEF --- (PRT) GHI --- (UNP) JKL --- (PRT)

These characters are unprotected, while the rest are protected.

### 3.2.9 Photo Macro Command

#### 3.2.9.1 General

The photo macro command facility provides the capability for a string of any presentation level characters and commands including transparent mode commands, to be stored within the terminal and to be subsequently executed via a single photo macro call. A photo macro name is one of the 96 characters (from 2/0 to 7/15), named PM0 through PM95.

#### 3.2.9.2 P-DEF MACRO

The P-DEF MACRO is specified by the parameter 4/0 following the P-MACRO code which is one of the C1 control codes. The P-DEF MACRO command is used to define a photo macro command. The character following the code is the name of the photo macro command. All character codes and transparent mode bit sequences are stored within the terminal under the specified photo macro name. Definition of the photo macro command terminates upon receipt of P-DEF MACRO, P-DEFP MACRO, P-DEFT MACRO and P-MACRO END commands.

Characters and commands following the photo macro name character are not executed at the time of reception. Neither the terminating code nor its preceding ESC code (in a 7-bit environment) is stored as part of the photo macro command.

Definition of a photo macro command replaces any previously existing photo macro command under the same name. If there are no valid characters or commands between the photo macro name and the terminating code (including its preceding ESC code), the photo macro command is deleted. All photo macros are simultaneously deleted with the P-RESET command or CS.

#### 3.2.9.3 P-DEFP MACRO

The P-DEFP MACRO is specified by the parameter 4/1 following the P-MACRO C1 control code.

This command is the same as the P-DEF MACRO command except that it simultaneously stores and executes the incoming characters that make up the macro.

#### 3.2.9.4 P-DEFT MACRO

The P-DEFT MACRO is specified by the parameter 4/2 following the P-MACRO C1 control code.

This command is used to define a transmit-macro. Transmit-macros, when called, are not executed, but are transmitted back to the host computer.

#### 3.2.9.5 P-MACRO END

The P-MACRO END is specified by the parameter 4/15 following the P-MACRO C1 control code.

This command terminates the current P-DEF MACRO, P-DEFP MACRO or P-DEFT MACRO operation.

### 3.3 Display control command set

#### 3.3.1 General

The display control command set provides control over display attributes, presentation level data format and display structure definition. Display control commands have effect on textual information, photographic information and geometric information.

The display control command set is one of the G sets. A display control command is composed of one single byte opcode and one or more operands if necessary. Each operand consists of one or more bytes of numeric data or bits combination.

There are three types of operands.

- i) Fixed format
- ii) Single-value format
- iii) Multi-value format

The fixed format operands consists of one or more bytes of numeric data or bits combination whose length and interpretation depends on the opcode with which they are used.

The single-value operands consists of one, two, three or four bytes of numeric data, as determined by the P-DOMAIN command described later. They are interpreted as unsigned integers composed of the sequence of concatenated bits taken consecutively (high order bit or b<sub>8</sub> to low order bit or b<sub>1</sub>) from the numeric data bytes as shown in Fig.3.1(1).

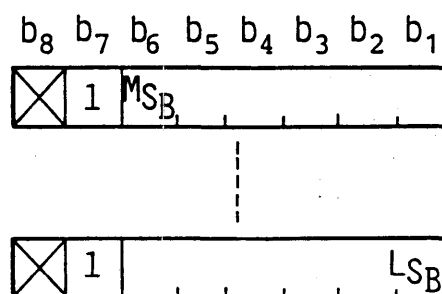


Fig.3.1(1) Single-Value Operand Format

The multi-value operands consist of one to eight bytes of numeric data, as determined by the P-DOMAIN command. These operands are used to specify coordinate information, when used to specify dimensions, or colour information, when used in conjunction with the SET LUT command described later.

All coordinate operands are interpreted as signed, two's complement numbers, with the MSB representing 0.5.

When the multi-value operand is used along with the SET LUT command, it specifies an unsigned colour value in the RGB (red-green-blue) colour system. The representation of the colour value within the multi-value operand is shown in Fig.3.1(2), where the colour value is given by an unsigned binary decimal. The colour value '0' indicates the lowest intensity, while '1' does the highest intensity.

Table 3.3 shows the types of operands used by each of the opcodes.

Entire display control command set is shown in Table 3.4.

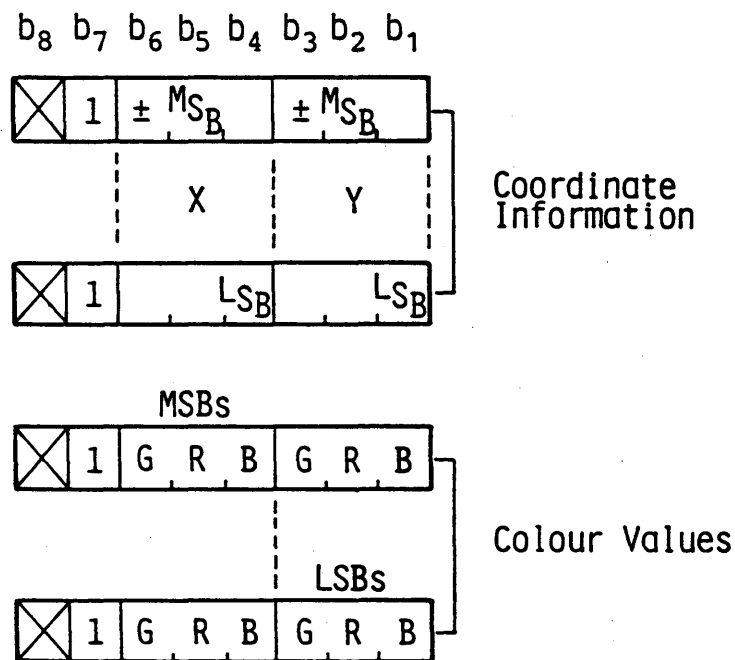


Fig.3.1(2) Multi-Value Operand Format

All coordinate operands are interpreted as signed, two's complement numbers, with the MSB representing 0.5.

When the multi-value operand is used along with the SET LUT command, it specifies an unsigned colour value in the RGB (red-green-blue) colour system. The representation of the colour value within the multi-value operand is shown in Fig.3.1(2), where the colour value is given by an unsigned binary decimal. The colour value '0' indicates the lowest intensity, while '1' does the highest intensity.

Table 3.3 shows the types of operands used by each of the opcodes.

Entire display control command set is shown in Table 3.4.

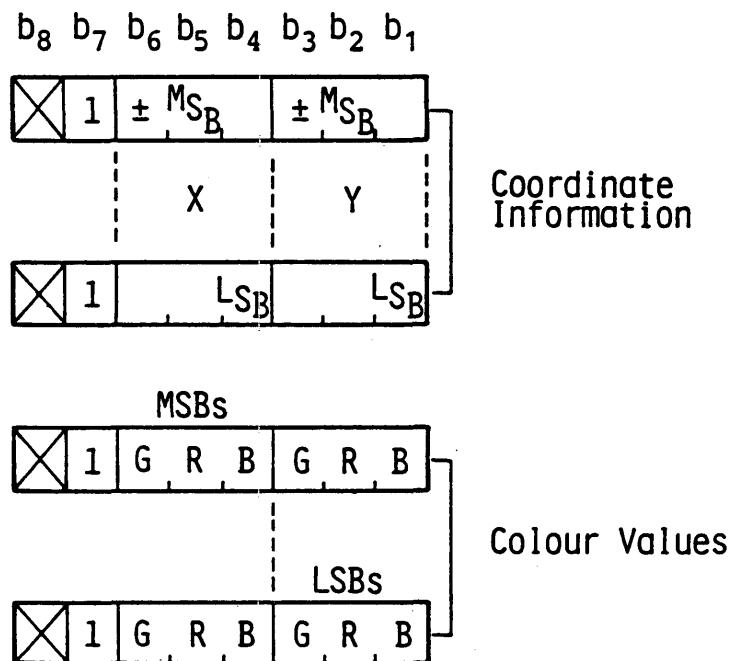


Fig.3.1(2) Multi-Value Operand Format

Table 3.4 Display Control Command Set

					10	11	12	13	14	15
				b <sub>7</sub>	0	0	1	1	1	1
				b <sub>6</sub>	1	1	0	0	1	1
				b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7
0	0	0	0	0			NUMERIC DATA			
0	0	0	1	1	P-RESET					
0	0	1	0	2	P-DOMAIN					
0	0	1	1	3	LOGICAL PEL					
0	1	0	0	4	DISPLAY MODE	P-TEST				
0	1	0	1	5	AREA					
0	1	1	0	6	SET FRAME					
0	1	1	1	7	ASSIGN FRAME					
1	0	0	0	8		RASTER				
1	0	0	1	9		HEADER RASTER				
1	0	1	0	10		SET LUT				
1	0	1	1	11						
1	1	0	0	12	P-WAIT	P-BLINK				
1	1	0	1	13						
1	1	1	0	14						
1	1	1	1	15						



### 3.3.2 Attribute control functions

#### 3.3.2.1 P-DOMAIN

The P-DOMAIN command defines the operand length for both character code mode commands (display control commands, PDI and MVI) and transparent mode commands (photographic commands) and the dimensionality. These parameters, once set, remain unchanged until altered by another P-DOMAIN command or reset by NSR or CS.

The P-DOMAIN command takes two bytes fixed format operands as shown in Fig.3.2. The first byte defines the character code commands operand length and the second does the transparent mode commands operand length. Bits of both bytes are interpreted as follows.

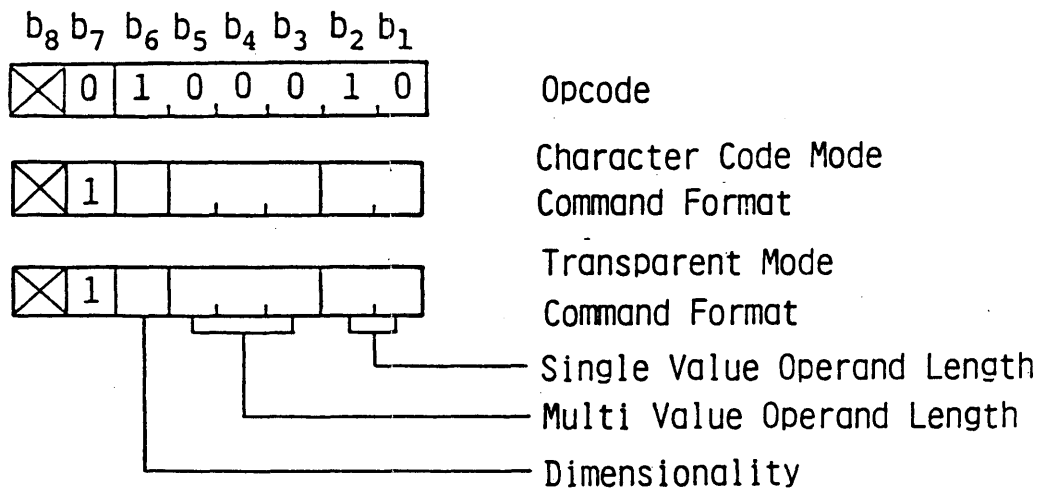


Fig.3.2 P-DOMAIN

#### i) Single-value operand length

b2	b1	single-value operand length (Bytes)
0	0	1 *
0	1	2
1	0	3
1	1	4

\* Default value for both character code commands and transparent mode commands.

(The single value transparent mode command operand format is shown in Fig.3.3(1).)

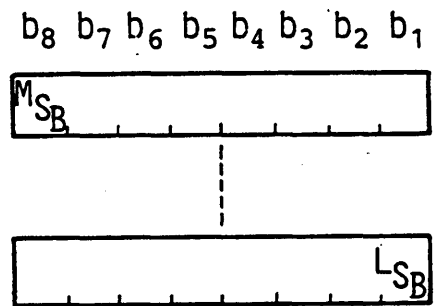


Fig.3.3(1) Single-Value Transparent Mode Command Operand

ii) Multi-value operand length

b5	b4	b3	Multi-value operand length (Bytes)	
0	0	0	1	
0	0	1	2	**
0	1	0	3	***
0	1	1	4	
1	0	0	5	
1	0	1	6	
1	1	0	7	
1	1	1	8	

\*\* Default value for transparent mode commands  
\*\*\* Default value for character code mode commands  
(The multi-value transparent mode command operand format is shown in Fig.3.3(2).)

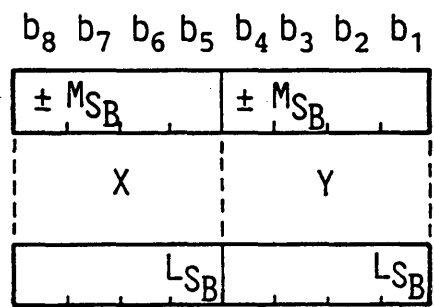


Fig.3.3(2) Multi-Value Transparent Mode Command Operand

### iii) Dimensionality

Bit 6 of each operand byte determines the dimensionality of the coordinate specification. 0 indicates two dimensional mode, and 1 indicates three dimensional mode, however, the definition of three dimensional mode is reserved for future standardization. Therefore, bit 6 should be always 0 until final standardization.

### iv) Operand length

If an operand following a character code mode command opcode is shorter than the length previously specified by the P-DOMAIN command (or the implicit length in the fixed format case), then the remainder of the operand is padded with zeros, unless otherwise indicated in the definition of the command. If an operand following a character code mode command opcode is longer than the length previously specified by the P-DOMAIN command (or the implicit length), it is taken as an indication to repeat the execution of the command with the subsequent numeric data taken as new operands, unless otherwise indicated.

#### 3.3.2.2 LOGICAL PEL

The LOGICAL PEL command defines the size of the logical pel and the colouring block. The LOGICAL PEL command takes two multi-value operands as shown in Fig.3.4. The first operand specifies the logical pel size. The width (dx) and height (dy) of a logical pel are interpreted as relative coordinate values.

The second multi-value operand specifies the colouring block size. The colouring block size is common for the foreground colour plane, the background colour plane and the display attribute plane, and is normally larger than the logical pel.

#### 3.3.2.3 Colour look up table controls

The SET LUT command and the P-BLINK command allow control over the colour look up table (LUT). The block colouring frame's LUT is commonly used for both foreground colours and background colours.

##### 3.3.2.3.1 SET LUT

The SET LUT command is used to download RGB colour values into the LUT. The SET LUT command takes one single-value operand followed by one multi-value operand as shown in Fig. 3.5. The single-value operand specifies the LUT entry which is left justified within the operand. The multi-value operand specifies the colour values for the entry assigned. If addi-

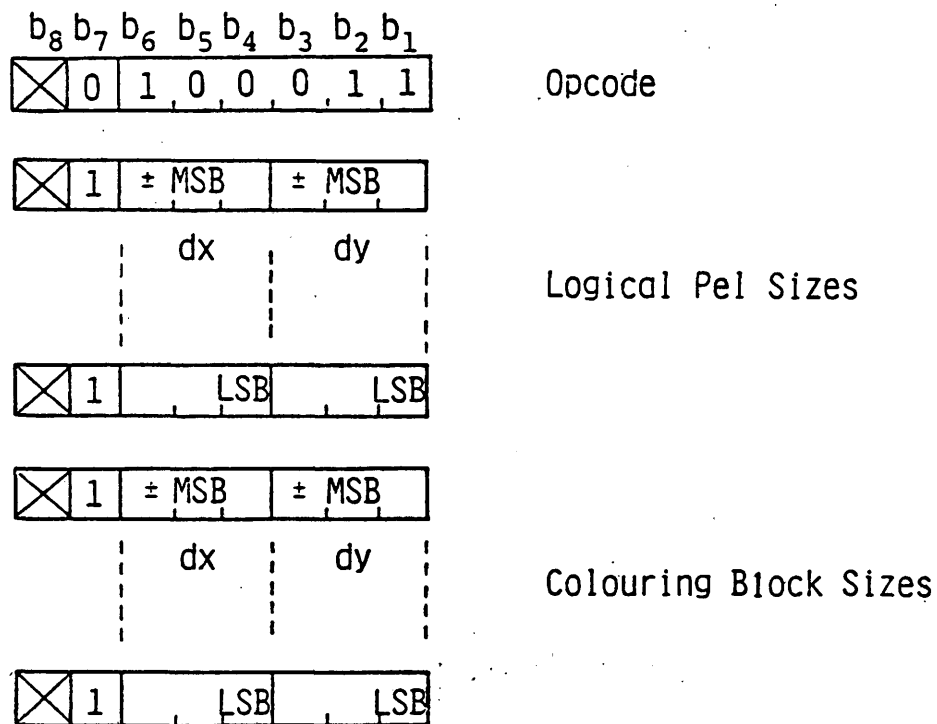


Fig.3.4 LOGICAL PEL

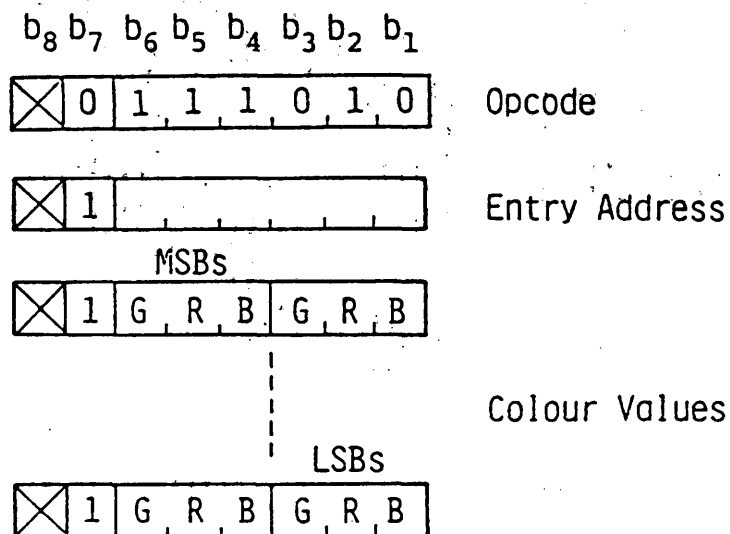


Fig.3.5 SET LUT

tional numeric data follows this operand, then the SET LUT command is implicitly repeated with the entry address of the LUT being automatically incremented. The incrementing process starts from the MSB side.

If there is no multi-value operand, each value of R, G and B is taken as 0, i.e. transparent.

The default contents of the LUT are shown in Table 3.5.

Table 3.5 Default Colour Look Up Table

Entry Address (RI,B,G,R)		Colour Values			Nominal Colour
		B	G	R	
0	0 0 0 0	0001	0001	0001	Black
1	0 0 0 1	0000	0000	1111	Red
2	0 0 1 0	0000	1111	0000	Green
3	0 0 1 1	0000	1111	1111	Yellow
4	0 1 0 0	1111	0000	0000	Blue
5	0 1 0 1	1111	0000	1111	Magenta
6	0 1 1 0	1111	1111	0000	Cyan
7	0 1 1 1	1111	1111	1111	White
8	1 0 0 0	0000	0000	0000	Transparent
9	1 0 0 1	0000	0000	0111	RI Red
10	1 0 1 0	0000	0111	0000	RI Breen
11	1 0 1 1	0000	0111	0111	RI Yellow
12	1 1 0 0	0111	0000	0000	RI Blue
13	1 1 0 1	0111	0000	0111	RI Magenta
14	1 1 1 0	0111	0111	0000	RI Cyan
15	1 1 1 1	0111	0111	0111	Grey

RI: Reduced Intensity

### 3.3.2.3.2 P-BLINK

The P-BLINK command causes a blink process at the LUT entry. The blink process periodically overwrites the contents of the specified entry (the "blink-from" colour) and substitutes another entry's contents in the LUT (the "blink-to" colour).

The P-BLINK command takes two single-value operands followed by three fixed format operands. The first single-value operand is the blink-from colour specification, specified as an LUT entry address. The second single-value operand specifies the blink-to colour.

The first fixed format operand following the second single-value operand specifies the ON interval which is a period of time the blink-to colour is activated. The second fixed format operand specifies the OFF interval which is a period of time the blink-from colour is activated. The third fixed format operand specifies the phase delay. The phase delay refers to the start of the ON interval of the most recently defined active blink process. These values are given in units of 1/10th of a second.

Fig.3.6 shows the P-BLINK command format.

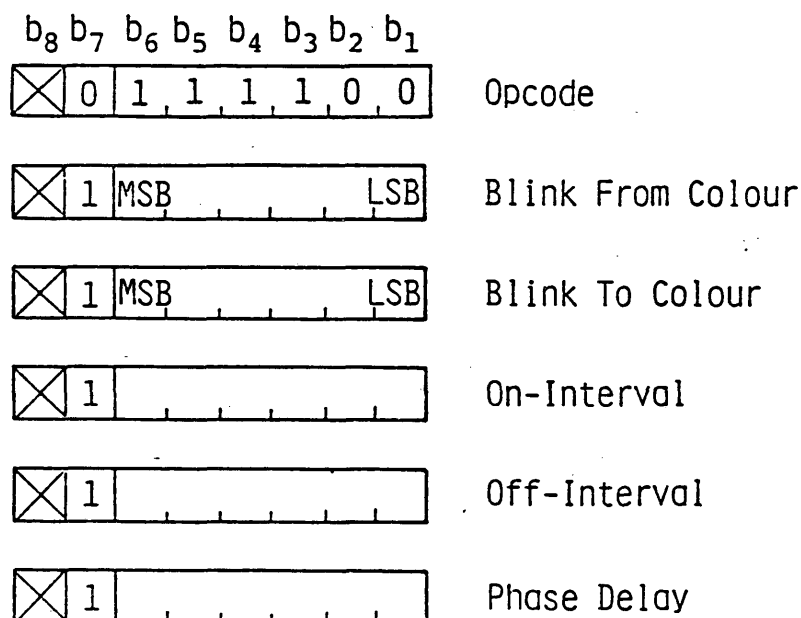


Fig.3.6 P-BLINK

### 3.3.2.4 Raster colour controls

The RASTER command and the HEADER RASTER command specify the raster colour and the header raster colour, respectively.

Both commands have the same format shown in Figs.3.7 and 3.8. Colour values are specified by a one multi-value operand. Once set, the raster colour or the header raster colour remains unchanged until another RASTER command, another HEADER RASTER command, or CS is executed. The default colour values of both the raster and the header raster colour are R = G = 0 and B = 0.0111 --- (= 1/2; Reduced blue colour).

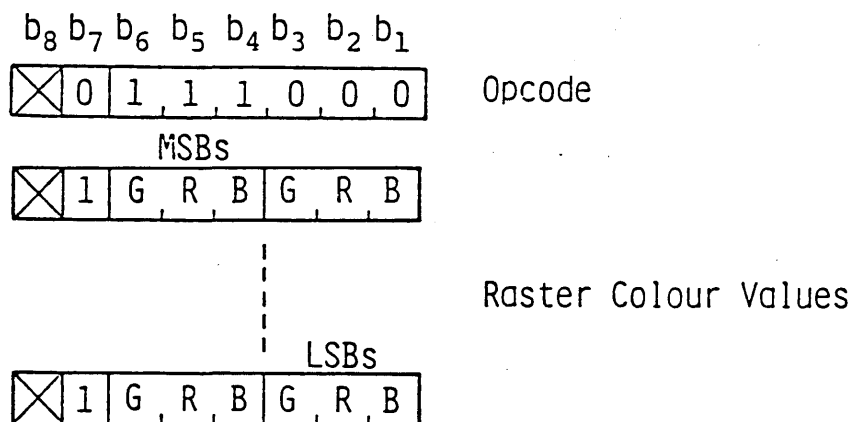


Fig.3.7 RASTER

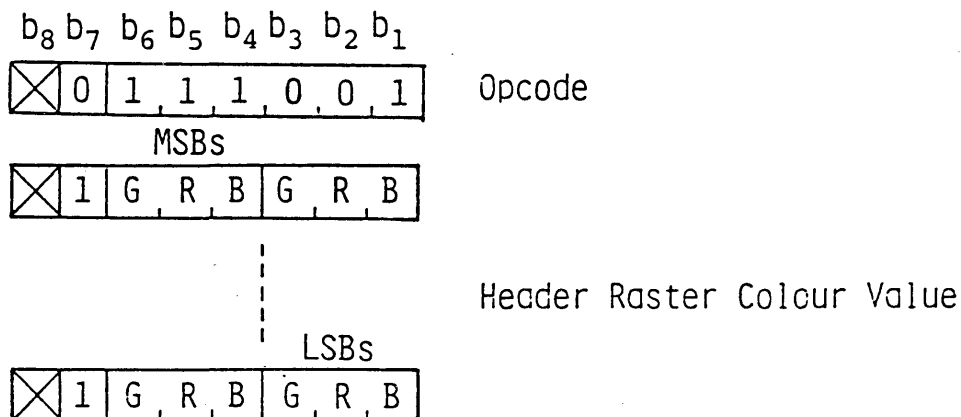


Fig.3.8 HEADER RASTER

### 3.3.2.5 P-RESET

The P-RESET command is used to selectively reset parameters as described below. It takes one single byte fixed format operand. (See Fig.3.9)

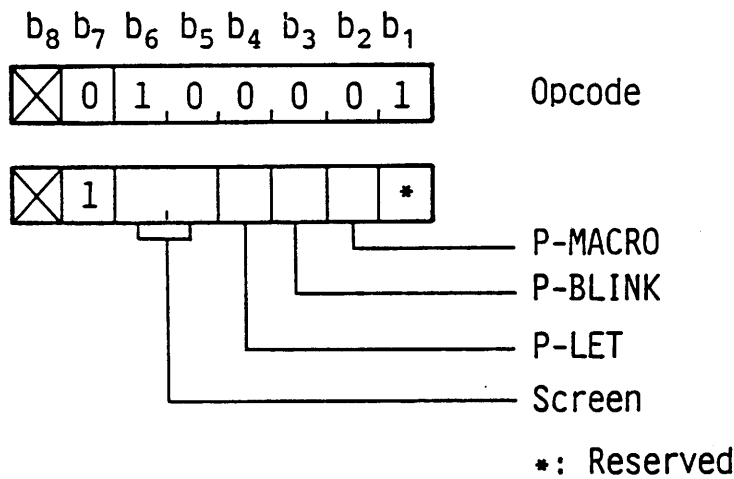


Fig.3.9 P-RESET

b6	b5	Screen
0	0	No operation
0	1	The entire pattern plane is reset to 0. The entire foreground colour is set to 7th colour of the LUT (default colour; white), while the entire background colour is set to 8th colour of the LUT (default colour; transparent)

When the screen is cleared (other than  $b_6 = b_5 = 0$ ), all flashing, conceal and protect attributes are cancelled.

b4	LUT
0	No operation
1	The LUT is set to default colours.
b3	P-BLINK
0	No operation
1	All blink processes are terminated.
b2	P-MACRO
0	No operation
1	All photo-MACROs are cancelled.



### 3.3.2.6 DISPLAY MODE

The DISPLAY MODE command specifies the display attributes. The DISPLAY MODE opcode takes one single byte fixed format operand whose bits (from b4 to b6) control individual attributes. (See Fig.3.10)

b6	Scroll
0	Scroll off
1	Scroll on

The default state is scroll off.

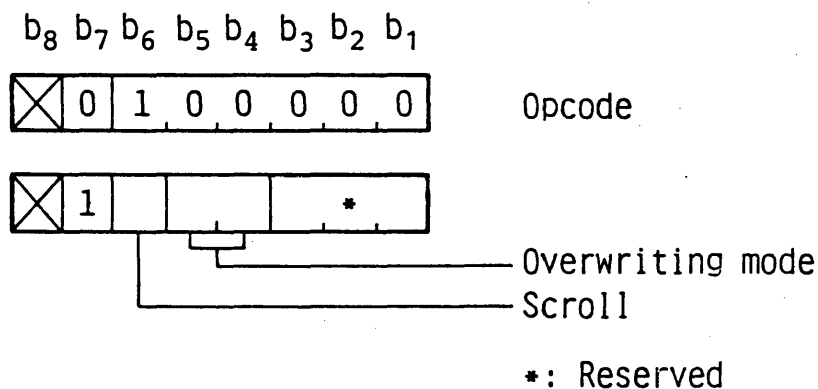


Fig.3.10 DISPLAY MODE

Operations of scroll on mode and scroll off mode are as follows.

i) Scroll off mode

- In case of a photographic command  
Data given are written to the point specified by the command.
- In case of text  
When the next character position exceeds the lower boundary of the screen, the next character will be displayed at the upper left corner of the screen.

ii) Scroll on mode

- In case of a photographic command  
When the scroll on mode is activated by the DISPLAY

MODE command, a register for storing a Y coordinate value (Y coordinate register) is first preset to 0.75. Upon receipt of new data, full contents of the screen are scrolled upward by the distance given as the difference between the contents of the Y coordinate register and the Y coordinate value indicated by the command, then new data is written at the position Y = 0. If the difference becomes 0.75, it is taken as 0. After that, the Y coordinate value indicated by the command is set to the Y coordinate register.

When a photographic command indicates to write data on a block by block basis rather than a line by line basis and the new field block position exceeds the lower boundary of the screen, the full contents of the screen are scrolled upward so that the bottom of the new field block meets the lower boundary of the screen.

• In case of text

When the next character position exceeds the boundary of the screen, the full contents of the screen are scrolled so that the next character can be displayed within the screen. Scrolling occurs in a direction perpendicular to the character path.

b5	b4	Overwriting mode.
0	0	New data for the pattern plane replaces the old one. $M_i \leftarrow D_i$ (replace mode)
0	1	New data and the old one is ORed and the result is written to the memory. $M_i \leftarrow M_i \quad \text{OR} \quad D_i$
1	0	New data and the old one is ANDed and the result is written to the memory. $M_i \leftarrow M_i \quad \text{AND} \quad D_i$
1	1	New data and the old one is Exclusive ORed and the result is written to the memory. $M_i \leftarrow M_i \quad \text{EOR} \quad D_i$

where  $M_i$  is the content of the pattern plane memory, and  $D_i$  is new data for it.

The default mode is the replace mode,

### 3.3.2.7 P-TEXT

The P-TEXT command is used to modify parameters which describe the text display manner. The opcode is followed by a two

byte fixed format operand and a multi-value operand. (See Fig.3.11).

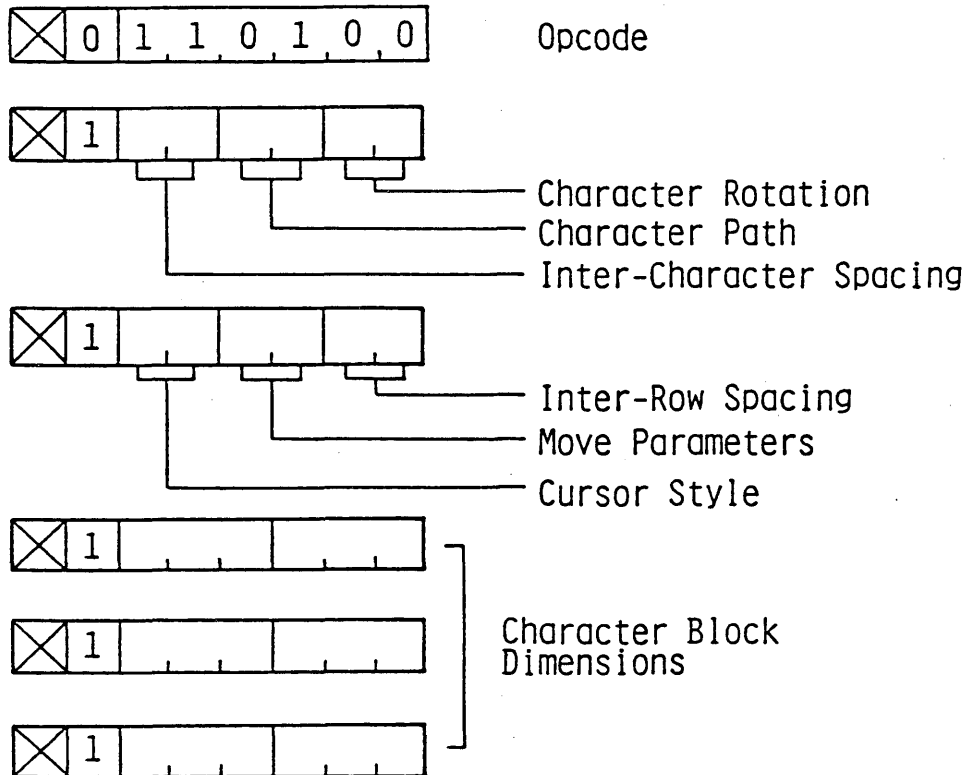


Fig.3.11 P-TEXT

The fixed format operand determines parameters as follows:  
**Inter-Character Spacing.** Bits b5 and b6 of byte 1 are used to determine the distance the cursor is moved (in multiples of the character field dimension lying parallel to the character path) after a character is displayed or after a SPACE or APB (backspace) or APF (horizontal tab) character is received.

b6	b5	Inter-character spacing
0	0	1 (default)
0	1	1.25
1	0	1.5
1	1	Proportional to the width of the character

The three fixed inter-character spacings (1, 1.25, and 1.5) are interpreted as multiplicative functions of the dimension of the

current character field lying parallel to the character path that are applied to movements of the cursor. In the proportionally spaced mode, the inter-character spacing is a variable that may be a function of the width of the actual pattern deposited as well as the current character size and font style. The proportional spacing algorithm is implementation dependent. However, each character shall be completely contained within the area defined by the current character field. This means that the exact number of characters per line is not known in proportional spacing mode, but it is at least as many characters per line as would be allowed by the current character field dimensions. The default inter-character spacing is a fixed space of 1.

Character Path Movement. Bits b3 and b4 of byte 1 determine the direction of the character path, that is, the direction in which the cursor is automatically advanced after a character is deposited. The character path is defined relative to horizontal within the unit screen and is independent of the character rotation. The default character path is right.

b4	b3	Character path movement
0	0	right (default)
0	1	left
1	0	up
1	1	down

Character Rotation. Bits b1 and b2 of byte 1 are used to specify character rotation.

b2	b1	Rotation
0	0	0° (default)
0	1	90°
1	0	180°
1	1	270°

Rotation causes the character field to rotate counter-clockwise about the character field origin. This rotation is measured relative to horizontal within the unit screen and is independent of the character path. The character field origin is the lower left corner of the character field at the default 0° rotation regardless of the sign of the character field dimensions dx and dy. All images within a character field are affected by rotation so that the relative position of the images within the character field is unchanged.

Cursor styles. Bits b5 and b6 of byte 2 are used to deter-

mine the display style of the cursor symbol.

b6	b5	Cursor style
0	0	underscore (default)
0	1	block
1	0	cross hair
1	1	custom

This cursor is located in the position in which the next character is to be deposited. The underscore cursor symbol is a single line the width of the current character field at the bottom of the character field. The block cursor symbol is a solid block or outline of block whose size is the size of the current character field. The cross-hair cursor symbol consists of a vertical line and a horizontal line that intersect at the centre of the character field and whose height and width are equal to the height and width of the current character field. The definition of the shape of the custom cursor symbol is implementation independent.

Move Attributes. Bits b3 and b4 of byte 2 are used to define the relationship between movement of the cursor and movement of the graphics drawing point.

b4	b3	Move attributes
0	0	The cursor and the drawing point move together
0	1	The cursor leads
1	0	The drawing point leads
1	1	The cursor and the drawing point move independently. (default)

If the cursor and the drawing point are set to move together (00), then whenever the cursor is moved (such as when characters are displayed) the graphic drawing point is moved with it, maintaining its alignment relative to the cursor. Correspondingly, whenever the drawing point is moved (such as with a geometric drawing primitive) the cursor is also moved so as to maintain its alignment relative to the drawing point.

If the cursor is defined as leading (01), then every time the cursor is moved the drawing point will move along with it but not vice versa,

If the drawing point is set to lead (10) the cursor then every time the drawing point is moved the cursor will move with it but not vice versa.

If the drawing point and the cursor are set to move independently (11), then movement of one will not affect the position of the other.

The alignment of the drawing point corresponds to the character field origin for the underscore cursor and block cursor, and the centre of the character field for the crosshair cursor and custom cursor.

The execution of a P-TEXT command shall cause alignment of the drawing point if the 'move together' or 'cursor leads' move attribute is in effect after execution. The execution of a P-TEXT command shall have no effect on the position of the character field origin.

Inter-Row Spacing. Bits b1 and b2 of byte 2 determine the inter-row spacing of characters, which defines the relative location of the cursor when it is advanced to a new line in a direction perpendicular ( $-90^\circ$ ) to the character path, either automatically as described below or by the APD (line feed) or APU (vertical tab) characters. Inter-row spacings are interpreted as multiplicative functions of the dimension of the character field that lies perpendicular to the character path.

b2	b1	Inter-row spacing
0	0	1 (default)
0	1	1.25
1	0	1.5
1	1	2

When using fixed or proportional inter-character spacing, if the character field origin is advanced along the character path such that any part of the full corresponding character field would exceed the edge of the unit screen, an automatic APR (carriage return) and APD (line feed) are executed. An inter-row spacing of 1, in which the character field on the current row abuts the character field of the previous row, is the default.

Character Block Dimensions. The multi-value operand specifies the width (dx) and the height (dy) of the character block which defines the 'normal' character size. (default values are  $dx = 0.0625$  and  $dy = 0.09375$ .) The character block size specified remains unchanged until another P-TEXT command, NSR or CS is received.

If the multi-value operand is omitted, the character block size remains unchanged. If more operands follow the specified opcodes, they are discarded.

If dx is negative, the character pixel patterns are reflect-

ed about the vertical centre axis of the character field. If  $dy$  is negative, the character pixel patterns are reflected about the horizontal centre axis of the character field.

### 3.3.2.8 P-WAIT

The P-WAIT command causes a specific time of delay in processing of text and commands following the command. The command format is shown in Fig.3.12. The P-WAIT opcode takes a single-value format operand which gives the time delay in units of 1/10th of a second.

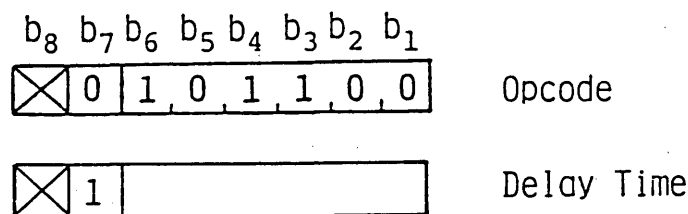


Fig.3.12 P-WAIT

### 3.3.3 Multi-frame control functions

#### 3.3.3.1 General

This standard provides the functionality which allows superimposed display of several frames and partition of the screen. The multi-frame display structure realizes display of textual information over pictorial information or animated pictorial information display over background pictorial information.

Screen partition provides various display structure: Different multi-frame structure and/or frame composition can be realized for different parts of the screen. This function enables to display different type of information such as photographic information and geometric information in a single screen with minimum raster memory requirement. For example, area A in Fig.3.13 has two photographic frames. Each frame consists of  $M$  bits per pixel, while the rest of the screen (area B) has three  $N$  bits frames. If  $2M$  equals to  $3N$ , both area A and B can be accommodated in the same raster memory.

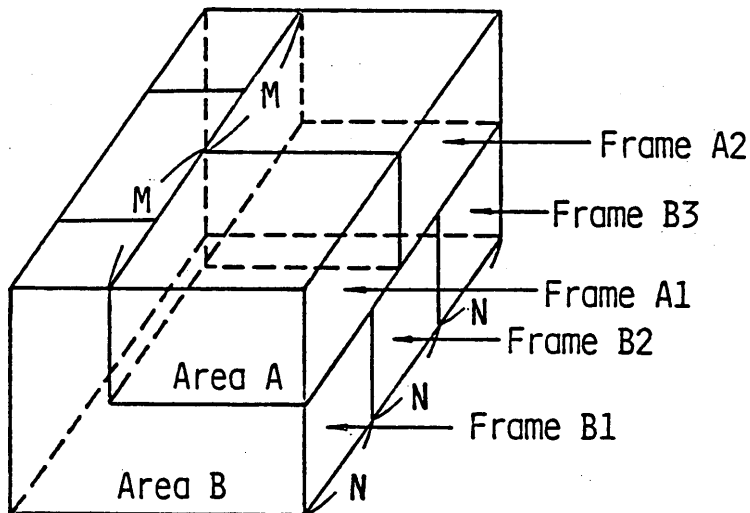


Fig.3.13 Screen Partition Concept

### 3.3.3.2 General flow of multi-frame display operation

The general flow for multi-frame display operations is shown in Fig.3.14.

At first, an active drawing area is defined. The following operations like bit assignment, frame selection and drawing commands are only effective to a defined area. Secondly, a multi-frame structure and bit-assignment for each frame is specified. Finally, a desired frame is selected. After that, general drawing commands follow.

### 3.3.3.3 AREA

The AREA command defines an active drawing area. Subsequent drawing commands are only effective in the defined area. The rest of the screen remains unchanged. For example, the scrolling of a picture can be executed within an area. There is only one active drawing area on the screen at a time. Area definition is cancelled, upon reception of a new AREA command.

An area is rectangular in shape, and defined by the originating position ( $x, y$ ) and width ( $dX$ ) and height ( $dY$ ). (See Fig. 3.15) The command format is shown in Fig.3.16. The AREA command itself has no effect on the displayed picture.



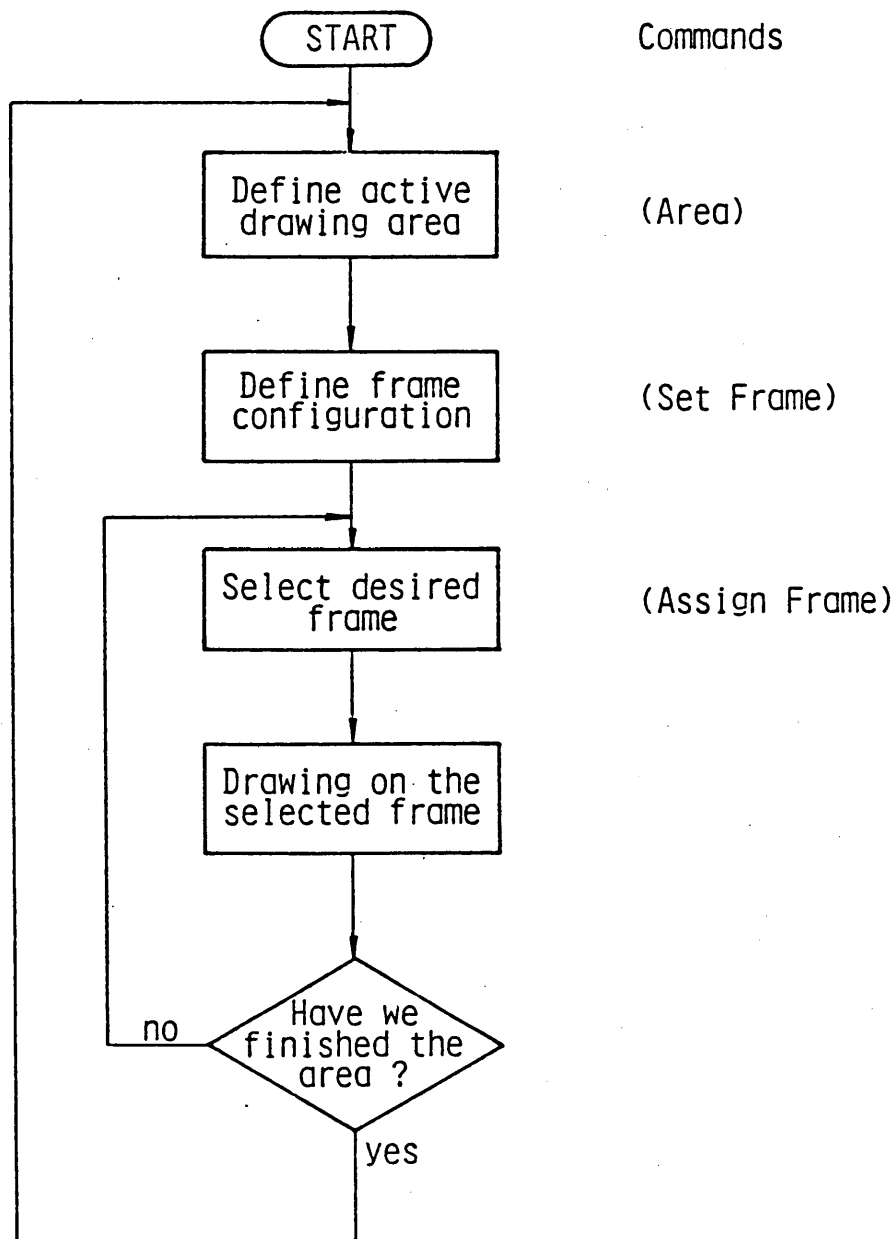


Fig.3.14 General Flow of Multi-Frame Display Operations

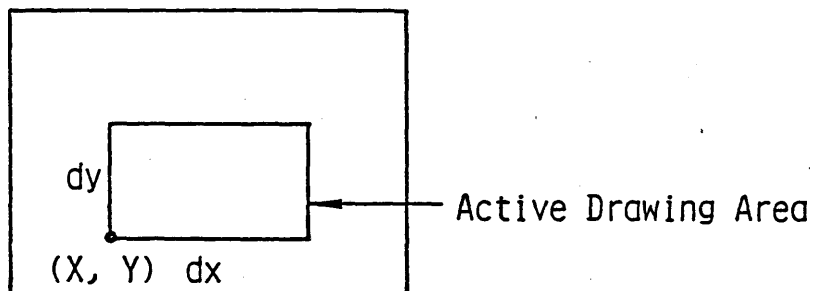


Fig.3.15 Active Drawing Area Definition

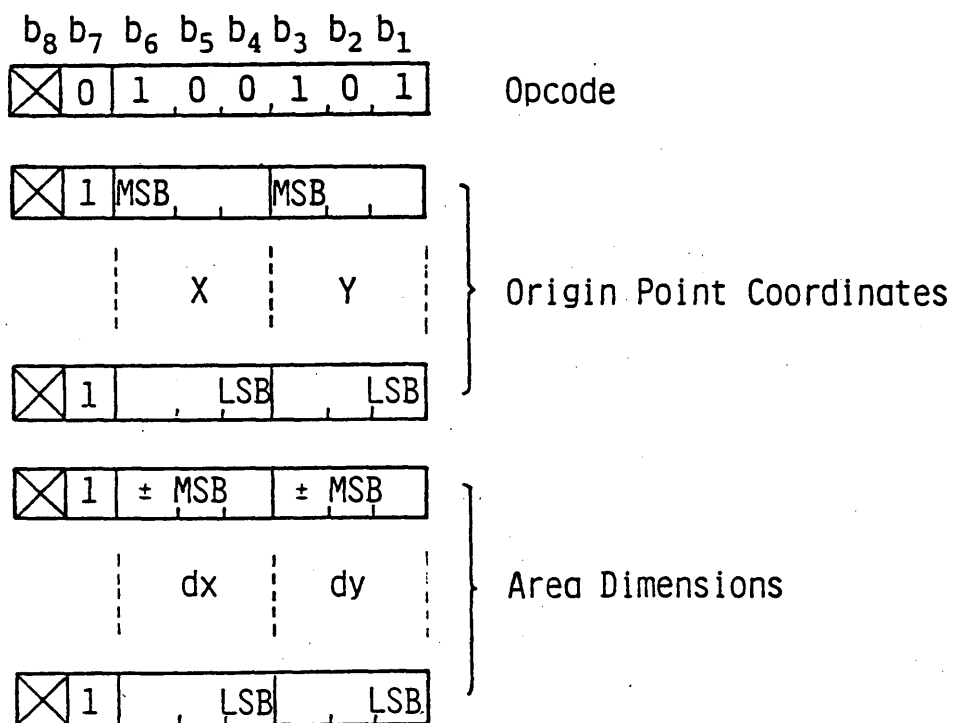


Fig.3.16 AREA

### 3.3.3.4 SET FRAME

The SET FRAME command specifies the raster memory configuration relevant to the defined area. The command format is

Opcode a1, b1, a2, b2, ... ,

as shown in Fig.3.17. Here, 'ai' is the frame index and 'bi' is the number of bits per raster memories allocated to the frame. The frame index indicates relative display priority. If ai is smaller than aj, the 'ai' frame has a higher display priority than the 'aj' frame. Any frame, however, has a higher display priority than the full screen background layer.

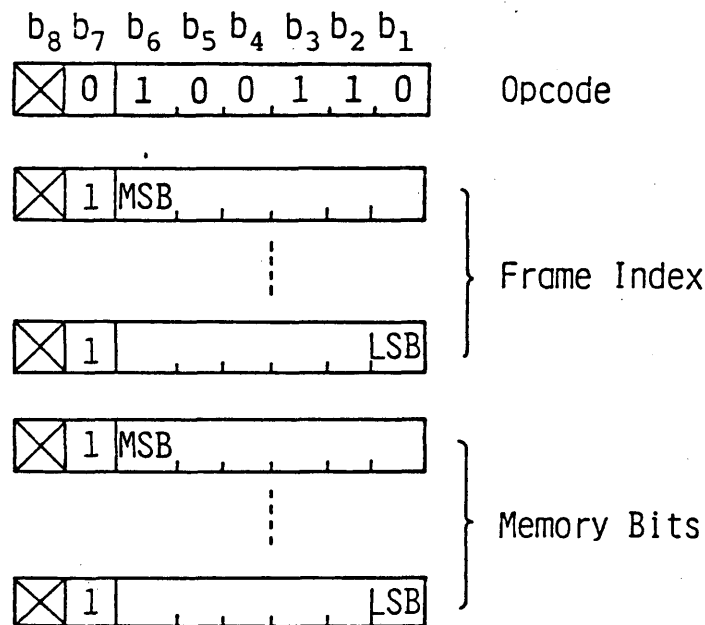


Fig.3.17 SET FRAME

### 3.3.3.5 ASSIGN FRAME

The ASSIGN FRAME command selects the frame on which succeeding drawings are made. The frame is assigned by the frame index stipulated by the SET FRAME command. The command format is

Opcoce ai,

as shown in Fig.3.18. Here, ai is the frame index.

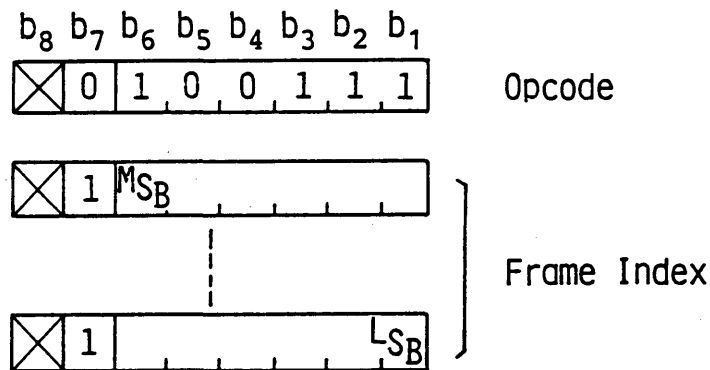


Fig.3.18 ASSIGN FRAME

Table 3.6 shows commands controlled by the ASSIGN FRAME command: Those commands have their effects only on the assigned frame.

Table 3.6 Commands Controlled by ASSIGN FRAME

CO control set	None
C1 control set	None
Display control command set	P-RESET, SET LUT and P-BLINK
Character sets	All
Mosaic sets	All
Dynamically Redefinable Character Sets	All
PDI set	All except RESET, DOMAIN, TEXTURE, SET COLOUR, WAIT, SELECT COLOUR and BLINK
MVI set	All
Photographic commands	All except P-DRCS 1 and P-DRCS 2

### 3.4 Character Sets

#### 3.4.1 Japanese-Kanji Character set

The Japanese-Kanji character set defines the total of 3657 characters including Japanese-Kanji characters, Japanese phonetic signs (Katakana and Hiragana), Roman characters, numerics and additional characters. Table 3.7 shows the Japanese-Kanji character set. Two byte codes are used for the Japanese-Kanji character set, where the first byte represents the 'Ku' number and the second byte represents the 'Ten' number. Characters 1 'Ku' 13 'Ten' through 18 'Ten' and 2 'Ku' 94 'Ten' are defined as non-spacing characters. Exact interpretation of 'Ku' or 'Ten' is the number specified by the byte plus 32.

The Japanese-Kanji characters can be displayed in normal size, double width, double height and double size.

#### 3.4.2 Primary Character set

The primary character set defines 52 Roman characters, 10 numerics, and 32 marks. Table 3.8 shows the primary character set. A primary character is represented by a single byte code. Any character size including medium size and small size are valid for the primary characters.

#### 3.4.3 Katakana Character set

The Katakana character set defines 63 Japanese phonetic signs. Table 3.9 shows the Katakana character set. A Katakana character is represented by a single byte code. Any character sizes are valid.

### 3.5 Mosaic sets

Two mosaic sets (Mosaic 1 set and Mosaic 2 set) are defined. Single byte codes are used for both Mosaic 1 and Mosaic 2 sets.

Both contiguous and separated shapes are provided for. In the underline mode activated by STL C1 control character, all characters from the mosaic sets are displayed in separated graphics representation.

#### 3.5.1 Mosaic 1 set

The Mosaic 1 set defines 59 graphic characters shown in Table 3.10.

#### 3.5.2 Mosaic 2 set

The Mosaic 2 set defines 94 graphic characters which are

Table 3.7 Japanese-Kanji set (1)

first byte							second byte							TEN	KU	
b7	b6	b5	b4	b3	b2	b1	b7	b6	b5	b4	b3	b2	b1			
0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	(sp)
0	1	0	0	0	1	0	2	1	1	1	1	1	1	2	2	◆
0	1	0	0	0	1	1	3	3	1	1	1	1	1	3	3	□
0	1	0	0	1	0	0	4	4	1	1	1	1	1	4	4	△
0	1	0	0	1	0	1	5	5	1	1	1	1	1	5	5	▽
0	1	0	0	1	1	0	6	6	1	1	1	1	1	6	6	※
0	1	0	0	1	1	1	7	7	1	1	1	1	1	7	7	?
0	1	0	1	0	0	0	8	8	1	1	1	1	1	8	8	!
0	1	0	1	0	0	1	9	9	1	1	1	1	1	9	9	→
0	1	0	1	0	1	0	10	10	1	1	1	1	1	10	10	←
0	1	0	1	0	1	1	11	11	1	1	1	1	1	11	11	↑
0	1	0	1	1	0	0	12	12	1	1	1	1	1	12	12	↓
0	1	0	1	1	0	1	13	13	1	1	1	1	1	13	13	=
0	1	0	1	1	1	0	14	14	1	1	1	1	1	14	14	〃
0	1	0	1	1	1	1	15	15	1	1	1	1	1	15	15	〃
0	1	1	0	0	0	0	16	16	1	1	1	1	1	16	16	〃
0	1	1	0	0	0	1	17	17	1	1	1	1	1	17	17	〃
0	1	1	0	0	1	0	18	18	1	1	1	1	1	18	18	〃
0	1	1	0	0	1	1	19	19	1	1	1	1	1	19	19	〃
0	1	1	0	1	0	0	20	20	1	1	1	1	1	20	20	〃
0	1	1	0	1	0	1	21	21	1	1	1	1	1	21	21	〃
0	1	1	0	1	1	0	22	22	1	1	1	1	1	22	22	〃
0	1	1	0	1	1	1	23	23	1	1	1	1	1	23	23	〃
0	1	1	1	0	0	0	24	24	1	1	1	1	1	24	24	〃
0	1	1	1	0	0	1	25	25	1	1	1	1	1	25	25	〃
0	1	1	1	0	1	0	26	26	1	1	1	1	1	26	26	〃
0	1	1	1	0	1	1	27	27	1	1	1	1	1	27	27	〃
0	1	1	1	1	0	0	28	28	1	1	1	1	1	28	28	〃
0	1	1	1	1	0	1	29	29	1	1	1	1	1	29	29	〃
0	1	1	1	1	1	0	30	30	1	1	1	1	1	30	30	〃
0	1	1	1	1	1	1	31	31	1	1	1	1	1	31	31	〃
1	1	0	0	0	0	0	32	32	1	1	1	1	1	32	32	〃
1	1	0	0	0	0	1	33	33	1	1	1	1	1	33	33	〃
1	1	0	0	0	1	0	34	34	1	1	1	1	1	34	34	〃
1	1	0	0	0	1	1	35	35	1	1	1	1	1	35	35	〃
1	1	0	0	1	0	0	36	36	1	1	1	1	1	36	36	〃
1	1	0	0	1	0	1	37	37	1	1	1	1	1	37	37	〃
1	1	0	0	1	1	0	38	38	1	1	1	1	1	38	38	〃
1	1	0	0	1	1	1	39	39	1	1	1	1	1	39	39	〃
1	1	0	1	0	0	0	40	40	1	1	1	1	1	40	40	〃
1	1	0	1	0	0	1	41	41	1	1	1	1	1	41	41	〃
1	1	0	1	0	1	0	42	42	1	1	1	1	1	42	42	〃
1	1	0	1	0	1	1	43	43	1	1	1	1	1	43	43	〃
1	1	0	1	1	0	0	44	44	1	1	1	1	1	44	44	〃
1	1	0	1	1	0	1	45	45	1	1	1	1	1	45	45	〃
1	1	0	1	1	1	0	46	46	1	1	1	1	1	46	46	〃
1	1	0	1	1	1	1	47	47	1	1	1	1	1	47	47	〃

first byte: "KU" number  
second byte: "TEN" number

Table 3.7 Japanese-Kanji set (2)

[illegible]

- 60 -

[illegible]



Table 3.7 Japanese-Kanji set (4)

first byte								second byte							
b7	b6	b5	b4	b3	b2	b1	TEN	b7	b6	b5	b4	b3	b2	b1	TEN
0	0	1	1	0	1	0	21	48	49	50	51	52	53	54	55
0	1	1	0	1	1	1	22	56	57	58	59	60	61	62	63
0	1	1	0	0	0	0	23	64	65	66	67	68	69	70	71
0	1	1	1	0	0	1	24	72	73	74	75	76	77	78	79
0	1	1	1	0	1	0	25	80	81	82	83	84	85	86	87
0	1	1	1	0	1	1	26	88	89	90	91	92	93	94	
0	1	1	1	1	0	0	27								
0	1	1	1	1	1	0	28								
0	1	1	1	1	1	1	29								
0	1	1	1	1	1	1	30								
1	0	0	0	0	0	0	31								
1	0	0	0	0	0	1	32								
1	0	0	0	0	0	1	33								
1	0	0	0	0	1	1	34								
1	0	0	0	1	0	0	35								
1	0	0	0	1	1	0	36								
1	0	0	0	1	1	1	37								
1	0	0	1	0	0	0	38								
1	0	0	1	0	0	1	39								
1	0	0	1	0	1	0	40								
1	0	0	1	0	1	1	41								
1	0	0	1	1	0	0	42								
1	0	0	1	1	1	0	43								
1	0	0	1	1	1	1	44								
1	0	0	1	1	1	1	45								
1	0	0	1	1	1	1	46								
1	0	0	1	1	1	1	47								
1	0	1	0	0	0	0	48								

Table 3.8 Primary Character Set

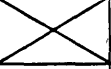








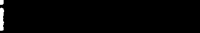


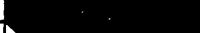


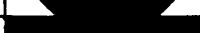



















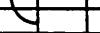





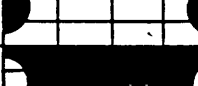

					10	11	12	13	14	15
				b <sub>7</sub>	0	0	1	1	1	1
				b <sub>6</sub>	1	1	0	0	1	1
				b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7
0	0	0	0	0		0	a	P	`	p
0	0	0	1	1	!	1	A	Q	a	q
0	0	1	0	2	"	2	B	R	b	r
0	0	1	1	3	#	3	C	S	c	s
0	1	0	0	4	\$	4	D	T	d	t
0	1	0	1	5	%	5	E	U	e	u
0	1	1	0	6	&	6	F	V	f	v
0	1	1	1	7	'	7	G	W	g	w
1	0	0	0	8	(	8	H	X	h	x
1	0	0	1	9	)	9	I	Y	i	y
1	0	1	0	10	*	:	J	Z	j	z
1	0	1	1	11	+	;	K	[	k	{
1	1	0	0	12	,	<	L	¥	l	
1	1	0	1	13	-	=	M	]	m	}
1	1	1	0	14	.	>	N	^	n	~
1	1	1	1	15	/	?	O	_	o	

Table 3.9 Katakana Character Set

					10	11	12	13	14	15
				b <sub>7</sub>	0	0	1	1	1	1
				b <sub>6</sub>	1	1	0	0	1	1
				b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7
0	0	0	0	0		ー	タ	ミ		
0	0	0	1	1	。	ア	チ	ム		
0	0	1	0	2	「	イ	ツ	メ		
0	0	1	1	3	」	ウ	テ	モ		
0	1	0	0	4	・	エ	ト	ヤ		
0	1	0	1	5	・	オ	ナ	ユ	Reserved	Reserved
0	1	1	0	6	ヲ	カ	ニ	ヨ		
0	1	1	1	7	フ	キ	ス	ラ		
1	0	0	0	8	イ	ク	ネ	リ		
1	0	0	1	9	ウ	ケ	ノ	ル		
1	0	1	0	10	エ	コ	ハ	レ		
1	0	1	1	11	オ	サ	ヒ	ロ		
1	1	0	0	12	ヤ	シ	フ	ワ		
1	1	0	1	13	ユ	ス	ヘ	ン		
1	1	1	0	14	ヨ	セ	ホ	ッ		
1	1	1	1	15	ッ	ソ	マ	。		

2(10)17 ~ 2(10)/15: Small size

Table 3.10 Mosaic 1 Set

					10	11	12	13	14	15	
					b <sub>7</sub>	0	0	1	1	1	1
					b <sub>6</sub>	1	1	0	0	1	1
					b <sub>5</sub>	0	1	0	1	0	1
b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>		2	3	4	5	6	7	
0	0	0	0	0							
0	0	0	1	1							
0	0	1	0	2							
0	1	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							

compatible with the second supplementary set of mosaic characters of Data Syntax II. (See Table 3.11)

### 3.6 Dynamically Redefinable Character Sets (DRCSs)

The Dynamically Redefinable Character Set (DRCS) provides a functionality whereby custom defined patterns can be downloaded and utilized as a G-set. Downloading is executed in the transparent mode through the Photo-DRCS 1 or Photo-DRCS 2 commands.

#### 3.6.1 DRCS 1

DRCS 1 patterns are identified by single byte codes. A DRCS code has a value in the range of 2/1 to 7/14. Pattern downloading is executed through the Photo-DRCS 1 command.

#### 3.6.2 DRCS 2

DRCS 2 patterns are identified by two bytes codes. A DRCS code has a value in the range 2/1 2/1 to 7/14 7/14. Pattern downloading is executed through the Photo-DRCS 2 command.

### 3.7 Macro set

A macro command consists of an arbitrary string of locally stored presentation level code that is identified by a code position (macro name) from the macro G-set. This name (from 2/0 to 7/15) thereafter acts as a substitute for the entire string of codes which is downloaded by the Photo-MACRO command.

### 3.8 Picture Description Instruction (PDI) set


The PDI set is identical to that of Data Syntax III except the TEXT command. (See Table 3.12) However, the programmable texture masks are not supported.

The following commands in the PDI set and the display control command set have identical effects on display attributes.

PDI set	Display control command set
RESET	P-RESET
DOMAIN	P-DOMAIN, LOGICAL PEL
SET COLOUR	SET LUT
WAIT	P-WAIT
BLINK	P-BLINK

The SELECT COLOUR command has the same effect as the C1 colour

Table 3.11 Mosaic 2 Set

					10	11	12	13	14	15	
					b <sub>7</sub>	0	0	1	1	1	1
					b <sub>6</sub>	1	1	0	0	1	1
					b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7	
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							

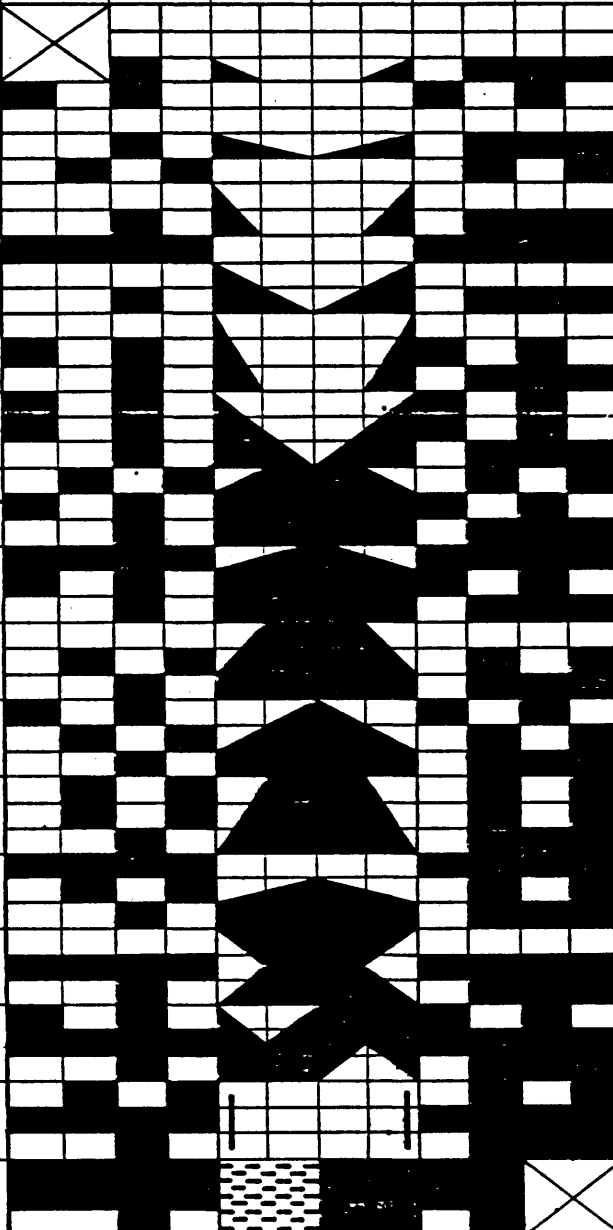


Table 3.12 PDI Set

					10	11	12	13	14	15
				b <sub>7</sub>	0	0	1	1	1	1
				b <sub>6</sub>	1	1	0	0	1	1
				b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7
0	0	0	0	0	RESET	RECT ANGLE (OUTLINED)	NUMERIC DATA			
0	0	0	1	1	DOMAIN	RECT ANGLE (FILLED)				
0	0	1	0	2		SET & RECT. (OUTLINED)				
0	0	1	1	3	TEXTURE	SET & RECT. (FILLD)				
0	1	0	0	4	POINT SET (ABS)	POLY GON (OUTLINED)				
0	1	0	1	5	POINT SET (REL)	PLOY GON (FILLED)				
0	1	1	0	6	POINT (ABS)	SET & POLY. (OUTLINED)				
0	1	1	1	7	POINT (REL)	SET & POLY. (FILLED)				
1	0	0	0	8	LINE (ABS)	FIELD				
1	0	0	1	9	LINE (REL)	INCREMENTAL POINT				
1	0	1	0	10	SET & LINE (ABS)	INCREMENTAL LINE				
1	0	1	1	11	SET & LINE (REL)	INCR. POLY. (FILLED)				
1	1	0	0	12	ARC (OUTLINED)	SET COLOUR				
1	1	0	1	13	ARC (FILLED)	CONTROL STATUS (WAIT)				
1	1	1	0	14	SET & ARC (OUTLINED)	SELECT COLOUR				
1	1	1	1	15	SET & ARC (FILLED)	BLINK				

control character have.

If there comes conflicting commands, the last received command has control over display attributes. In other words, display attributes are common for both PDI commands and photographic commands.

### 3.9 Moving Instruction (MVI) set

MOVE command (See 3.9.5.) causes the frame specified by ASSIGN FRAME to be moved virtually with reference to the other frames. (See Fig.3.19)

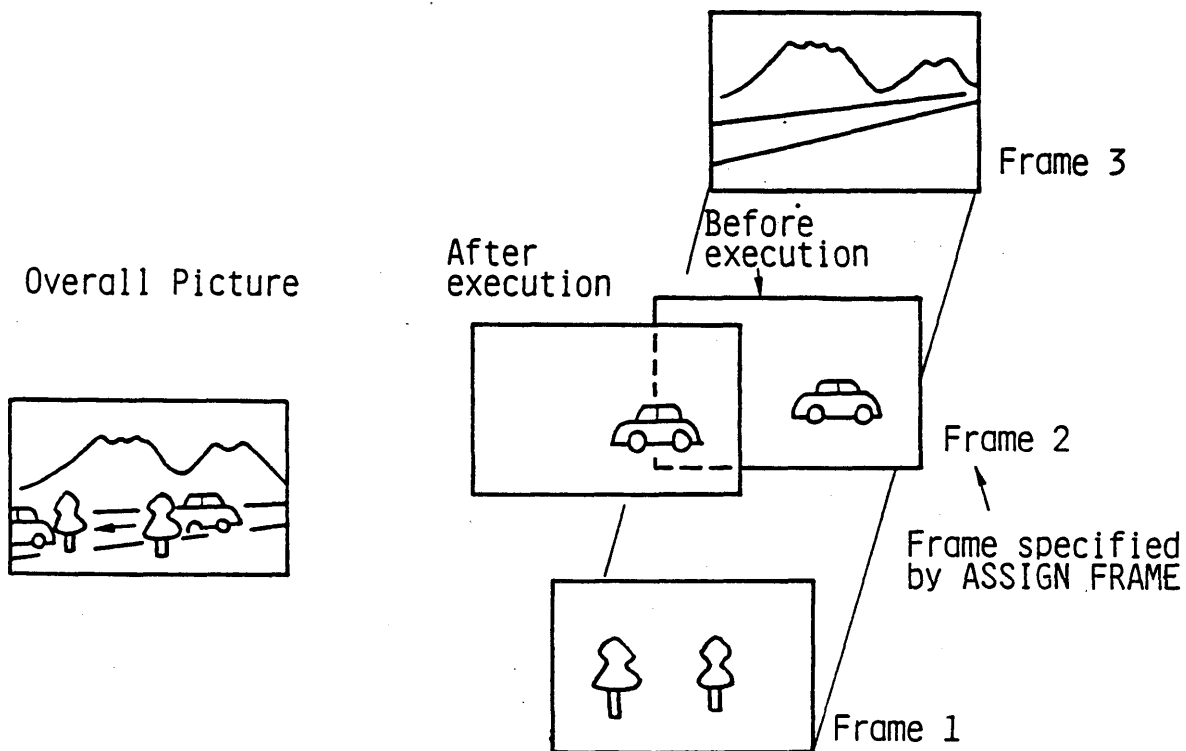


Fig.3.19 Move Instruction Examples

By the use of this command and the multi frame structure, animated picture presentation is available.

A moving instruction (MVI) is composed of an opcode and operands, being similar to PDIs. (Some instructions have no operands.)

There are single-value operands, multi-value operands and fixed format operands. The operand byte length of a single



value operand and a multi value operand are specified by the P-DOMAIN command in the display control command set.

The MVI set is shown in Table 3.13.

Table 3.13 MVI Set

					10	11	12	13	14	15
				b <sub>7</sub>	0	0	1	1	1	1
				b <sub>6</sub>	1	1	0	0	1	1
				b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7
0	0	0	0	0	MVI START		NUMERIC DATA			
0	0	0	1	1	MVI END					
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		MOVE				
1	0	1	1	11						
1	1	0	0	12		REPEAT START				
1	1	0	1	13		REPEAT END				
1	1	1	0	14						
1	1	1	1	15						

### 3.9.1 MVI START

MVI START indicates that all characters between MVI START and MVI END should be stored in the MVI buffer memory. (See Fig.3.20.) MVI START has no operand.

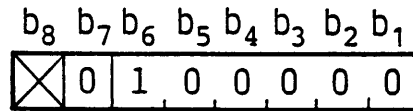


Fig.3.20 MVI START

### 3.9.2 MVI END

MVI END terminates storing commands, then activates the execution of the stored commands. (See Fig.3.21.) MVI END has no operand. When all stored characters are executed, the MVI buffer is cleared.

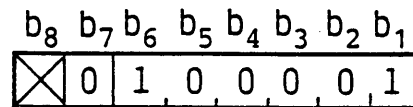


Fig.3.21 MVI END

### 3.9.3 REPEAT START

REPEAT START indicates that all presentation level characters between REPEAT START and REPEAT END are repeatedly executed. The number of repetition is specified by the REPEAT START operand. (See Fig.3.22.)

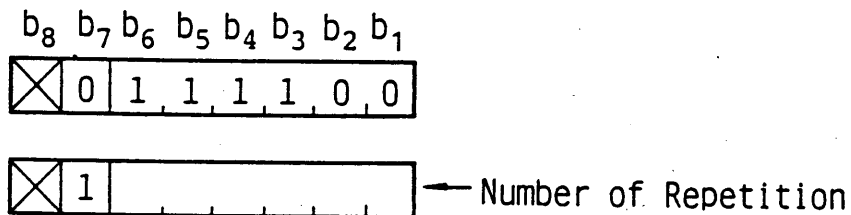


Fig.3.22 REPEAT START

### 3.9.4 REPEAT END

REPEAT END terminates the character strings to be repeated. REPEAT END has no operand. (See Fig.3.23.)

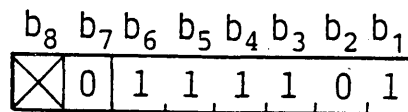


Fig.3.23 REPEAT END

### 3.9.5 MOVE

MOVE takes two multi-value operands and a following single-value operand. (See Fig.3.24.) The first multi-value operand specifies the originating point coordinates on the frame assigned by ASSIGN FRAME. The second multi-value operand specifies the terminating point coordinates. The specified frame moves from the originating point to the terminating point. The single-value operand immediately following the second multi-value operand indicates time period in the unit of one tenth second for which the frame moves the originating point to the terminating point.

The originating point coordinates represent the lower left corner coordinates of the frame specified by ASSIGN FRAME with respect to the unit screen. The terminating point coordinates are specified in the relative coordinates with reference to the originating point coordinates. For movement description, the unit screen boundary is expanded to -1.0 for both the X direction and Y direction.

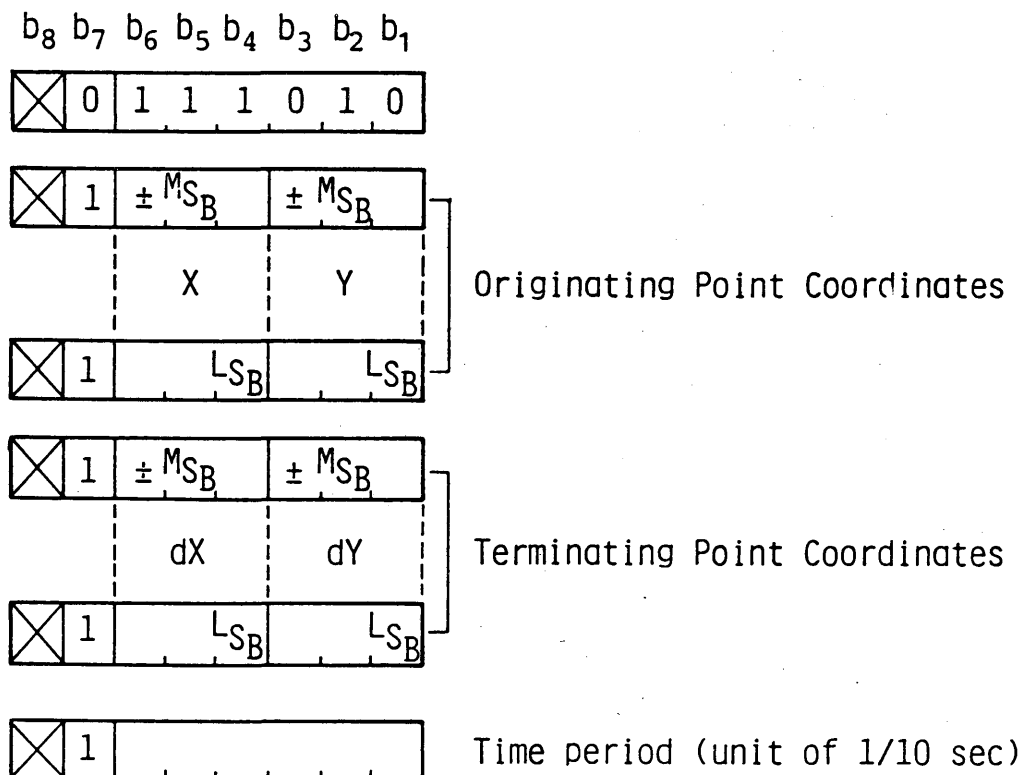


Fig.3.24 MOVE

#### 4. Coding in the transparent mode

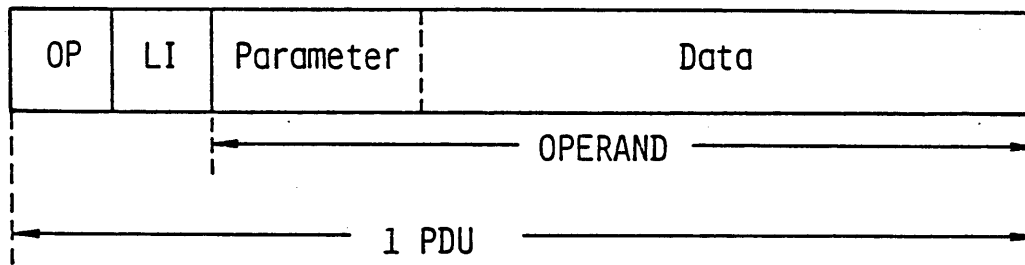
##### 4.1 Photographic Data Unit (PDU)

A photographic data unit (PDU) is composed of an opcode with a length indicator (LI), followed by zero, one or more operands, each of which consists of one or more octets of bit sequences.

The use of all octet patterns is allowed in the operands bit sequences, which results in efficient expression of arbitrary data. The PDU format is shown in Fig.4.1.

The opcode consists of a single octet which indicates the meaning of photographic data contained in the PDU.

The LI consists of one or more octet. The value of an LI is a binary number that represents the total length of operands following the LI field in octet. If the decimal expression of the second octet is 255, then the next two octets (the third and the fourth) indicate the real length. Since the LI simply indicates the data length, it is not shown in the figure of the PDU format throughout this section.



OP: Opcode

LI: Length Indicator

Fig.4.1 PDU Format

One or more octets of parameters are located at the leading part of an operand field. Parameters include the drawing point coordinates where the photographic data should be displayed, and/or the packing format which indicates the way in which photographic data are arranged. Photographic data expressed on a dot-by-dot basis are contained in the remaining part of the operand field.

Here, the dot corresponds to the logical pel or the colouring block in the unit screen. The pel position or the pel coor-

dinate is defined as one of the four corner positions of the logical pel, according to the signs of the width ( $dx_0$ ) and height ( $dy_0$ ) of the logical pel. The colouring block position is defined in the same manner as the pel position provided that the word 'logical pel' is replaced by the 'colouring block'.

There are three types of parameter operands: fixed format, single-value and multi-value. The length of single-value or multi-value parameter operands are determined by the P-DOMAIN command.

PDU opcodes are shown in Table 4.1. Blanks in the table are reserved for opcodes for dot-by-dot colouring (e.g. PCM, DPCM and transform coding), audio, telesoftware, etc..

#### 4.2 LINE DOT PATTERN

The LINE DOT PATTERN PDU format is shown in Fig.4.2. Photographic data express the values of logical pels, 1s and 0s, which should be written along a horizontal line on the pattern plane.

A single-value parameter operand indicates the absolute coordinate where the data should be written.

Correspondence between the photographic data and logical pel positions in the pattern plane is shown in Fig.4.3.

The first bit  $P_0$  is deposited at the leftmost logical pel position, next bit  $P_1$  to the right next pel position. When the next bit is received, the pel position moves to the right by one pel's size. If the bit  $P_n$  corresponds to the rightmost logical pel, the pel position corresponding to the next bit  $P_{n+1}$  should be moved at the leftmost pel position a logical pel height below.

If a bit corresponds to the lower right corner pel position, and if there are still remaining bits, they are discarded.

#### 4.3 LINE DOT PATTERN COMPRESSED

This PDU format is shown in Fig.4.4. Photographic data express the run length coded patterns of the logical pel values which should be written in the same manner as the LINE DOT PATTERN.

The single-value parameter operand indicates the absolute coordinate Y where the data is written.

Data compression is achieved by using the Modified Huffman (MH) run length coding technique. The coding process begins with 0's run, then alternately continues for 1's runs and 0's runs. The last run in a line is coded as an end of line code (EOL) regardless of its run length, since the last run continues to the end of line.

Bit sequence following EOL should be decoded for logical

Table 4.1 PDU Opcodes

				b <sub>8</sub>	0	0	0	0	
				b <sub>7</sub>	0	0	1	1	
				b <sub>6</sub>	1	1	0	0	
				b <sub>5</sub>	0	1	0	1	
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6 ~ 15
0	0	0	0	0	LINE DOT PATTERN (comp.)	LINE DOT PATTERN	COLOURING BLOCK (comp.)	COLOURING BLOCK	<ul style="list-style-type: none"> <li>• Telesoftware</li> <li>• Audio Data</li> </ul> # Dot by Dot Colouring with Redundancy Reduction  # Dot by Dot Colouring without Redundancy Reduction
0	0	0	1	1					
0	0	1	0	2		FIELD DOT PATTERN	FIELD COLOURING BLOCK (comp.)	FIELD COLOURING BLOCK	
0	0	1	1	3					
0	1	0	0	4			FREE FORMAT COLOURING BLOCK		
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	PHOTO DRCS1				
1	1	0	1	13	PHOTO DRCS2				
1	1	1	0	14					
1	1	1	1	15					

# Under Study

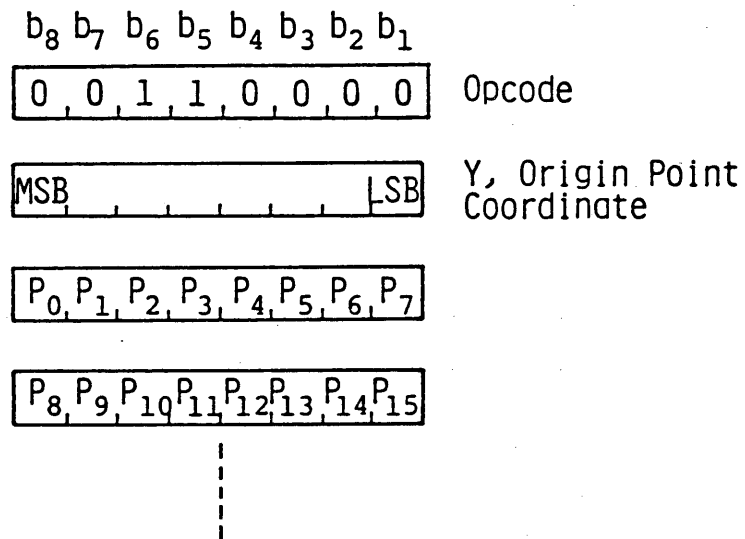


Fig.4.2 LINE DOT PATTERN

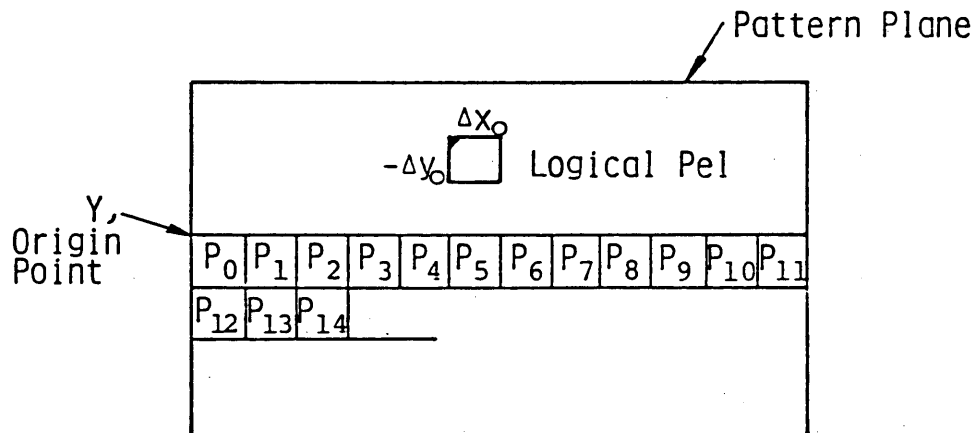


Fig.4.3 Correspondence between the Photographic Data and Logical Pel Positions in the Pattern Plane

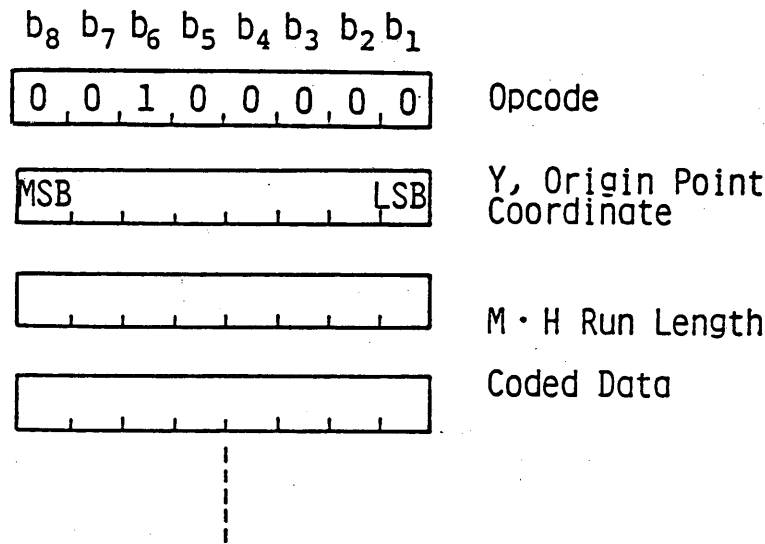


Fig.4.4 LINE DOT PATTERN COMPRESSED

pels one logical pel height below. The coding process starts with 0's run and continues in the same manner as the first line.

The MH code table is shown in Tables 4.2 and 4.3. A coding example is shown in Fig.4.5.

If more data are received after the pixel position reaches the lower screen boundary, they are discarded.

#### 4.4 FIELD DOT PATTERN

The FIELD DOT PATTERN PDU format is shown in Fig.4.6. Photographic data express the pel values in a rectangular field on the pattern plane.

There are two multi-value parameters operands which define the rectangular field location and dimensions in which the photographic data should be deposited.

The first multi-value operand specifies the initial logical pel position in absolute coordinates. The width and height (dx, dy) are given by the second multi-value operand. The initial pel position may be one of four vertices of the field since the width and height may be positive or negative.

Correspondence between the data and the logical pels in the pattern plane is shown in Fig.4.7. After a bit in the photographic data sequence is deposited at the current logical pel, the pel position is automatically moved in the x direction a distance equal to the width (dx<sub>0</sub>) of the logical pel. The



Table 4.2 MH Run Length Code Table (Part 1: 0's run)

	Terminating Code		Make up Code
0	00001	32	1110
1	0100	64	11011
2	0111	96	111100
3	1010	128	111110
4	1100	160	1111110
5	00100	192	00111110
6	00110	224	00111111
7	01100	256	01010110
8	01101	288	01010111
9	000100	320	01011110
10	000110	352	01011111
11	000111	384	10111110
12	001010	416	10111111
13	001110	448	11111110
14	010100	480	11111111
15	10110		
16	11010		
17	0000010		
18	0000011		
19	0001011		
20	0010110		
21	0010111		
22	0011110		
23	010110		
24	101110		
25	0101010		
26	0101110		
27	00010100		
28	1011110		
29	00010101		
30	1111010	EOL	100
31	1111011	Reserved	000000xx---

Table 4.3 MH Run Length Code Table (Part 2: 1's run)

	Terminating Code		Make Up Code
0	00000100	32	101100
1	01	64	10101010
2	11	96	10101011
3	100	128	10101100
4	0001	160	10101101
5	00100	192	10101110
6	10100	224	10101111
7	001010	256	10110100
8	001100	288	10110101
9	0000100	320	10110110
10	0011111	352	10110111
11	1010100	384	10111100
12	00000101	416	10111101
13	00000110	448	10111110
14	00000111	480	10111111
15	00001010		
16	001110		
17	00001011		
18	00001100		
19	00001101		
20	00001110		
21	00001111		
22	00101100		
23	00101101		
24	00101110		
25	00101111		
26	00110100		
27	00110101		
28	00110110		
29	00110111		
30	00111100	EOL	101110
31	00111101	Reserved	000000xx----

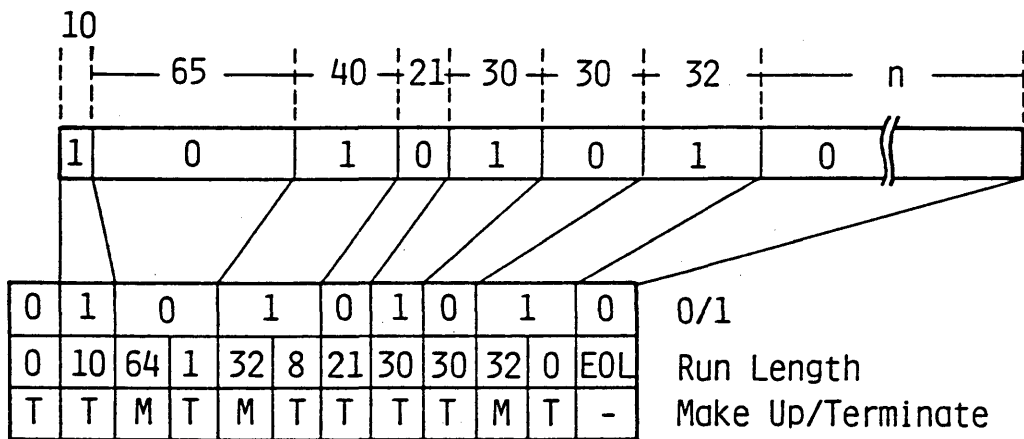


Fig.4.5 A Coding Example of LINE DOT PATTERN COMPRESSED

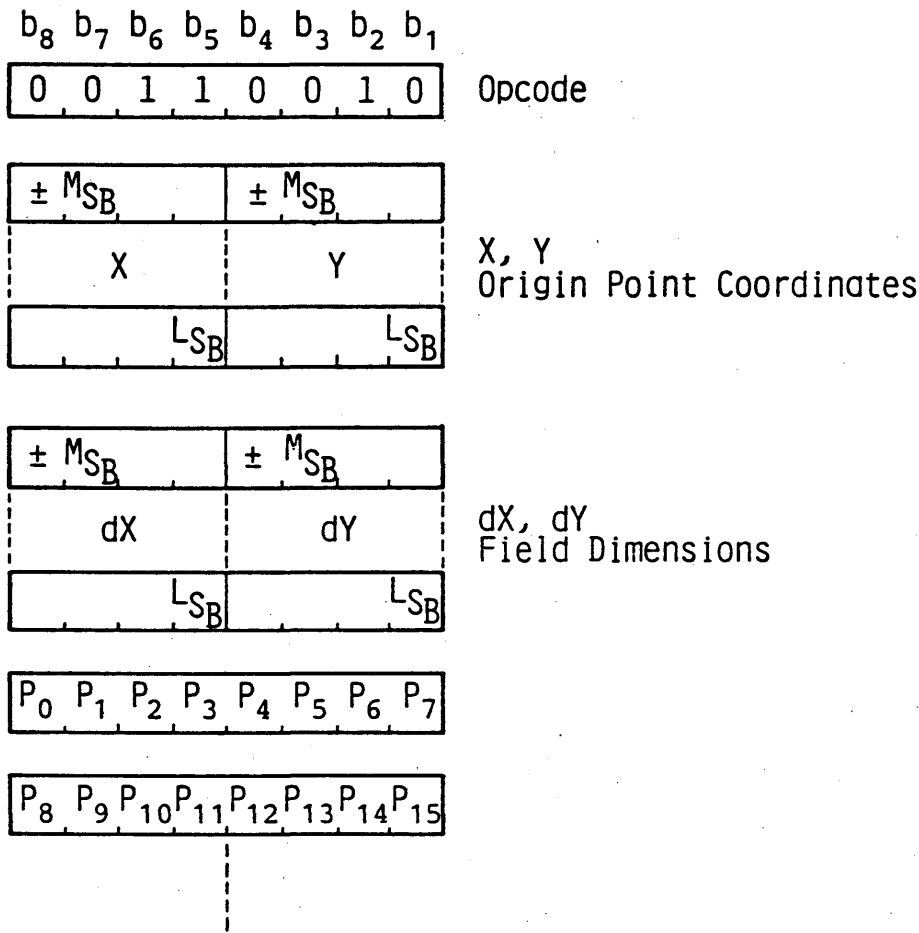


Fig.4.6 FIELD DOT PATTERN

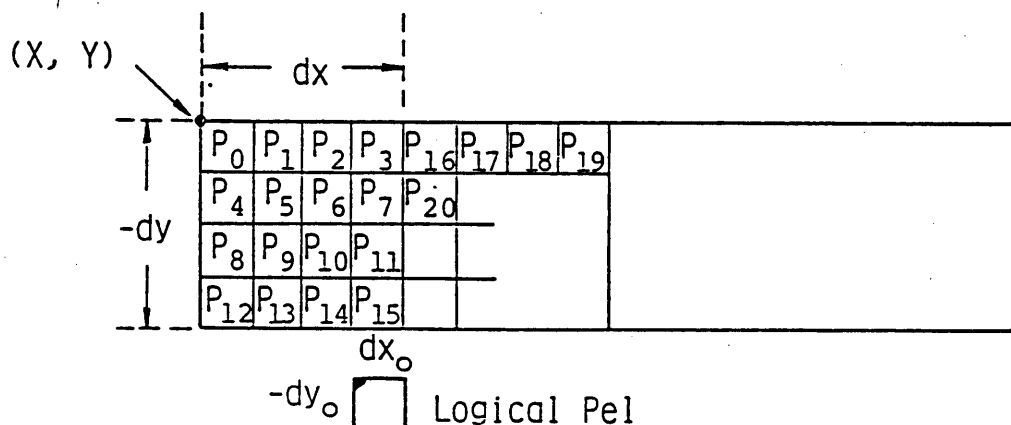


Fig.4.7 Correspondence between the Photographic Data and Logical Pel Positions

next bit is deposited at this position. If the field dimension  $dx$  is positive, position moves to the right and if  $dx$  is negative, position moves to the left.

If the pel position reaches or exceeds a vertical side in the field,  $x$  coordinate of the pel is moved to the other side of the field and the  $y$  coordinate is moved by the height ( $dy_0$ ) of the logical pel. If the field dimension  $dy$  is positive, the pel position moves up, and if  $dy$  is negative, the position moves down.

If the pel position reaches or exceeds a horizontal side of the rectangle. and if there are still remaining bits, they should be interpreted as if there proceeds the same opcode with field parameter operands  $(X + dx, Y)$ ,  $(dx, dy)$ .

#### 4.5 COLOURING BLOCK

The COLOURING BLOCK PDU format is shown in Fig.4.8. Photographic data express the colour and attributes of the colouring blocks in a horizontal line on the FG, BG and display attribute planes.

The first fixed format parameter operand indicates the colour and attribute flags. If the flag of the FG colour, BG colour or display attribute is '1', the PDU contains the corresponding data.

The second parameter, a single-value coperand, indicates the absolute coordinate  $Y$  where the attribute data decoding is initiated.

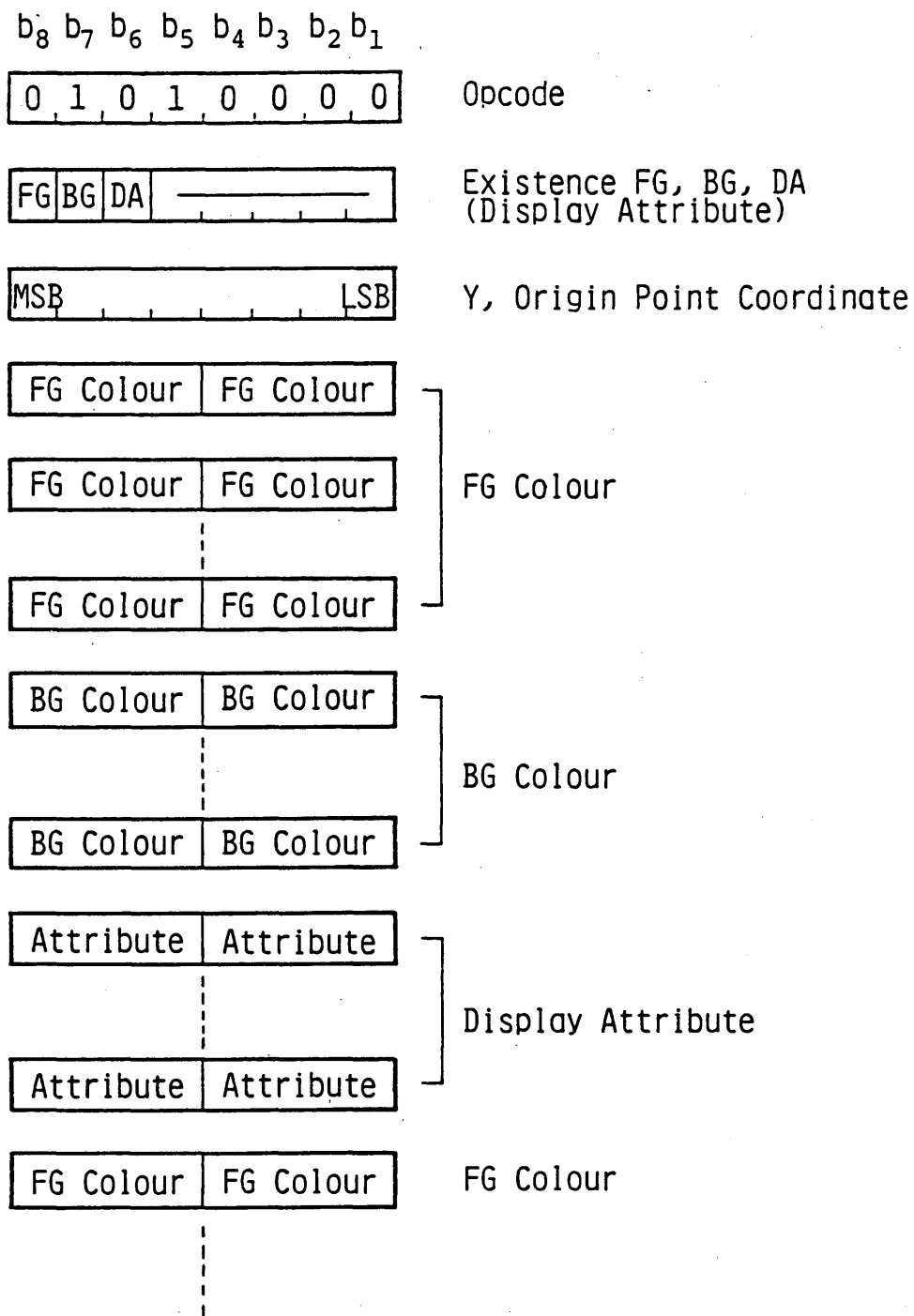










Fig.4.8 COLOURING BLOCK

Photographic data are composed of three successive groupings of bit codes which represent FG data, BG data and display attributes. A four-bit for FG data or BG data is an entry address of the LUT. Four-bit display attributes are used for flashing controls, and their meaning is shown in Table 4.4. When the first operand indicates all flags are 1s, FG colour data for the 1st line come first, BG colour data for the same line come next and then display attribute data for the same line follow. If there remain more data, they are interpreted as the data for colouring blocks in a line which is one colouring block height below. The remaining data are interpreted in the same manner as the first line.

Table 4.4 Display Attribute Code

Code	Flashing	Phase	Pattern
0000	ON	2	
0001	ON	3	
0010	ON	3	
0011	ON	3	
0100	ON	3	
0101	ON	3	
0110	ON	3	
0111	ON	2	
1111	OFF	—	—

Any data boundary in operand bytes (between FG colour data and BG colour data, between BG colour and display attributes, and between a line and a line below) keeps byte boundary, i.e., if last data occupy only a portion of a byte, remaining bits in the byte are stuffed with 0 bits.

If operand data overrun the lowest block boundary, they are ignored.

#### 4.6 COLOURING BLOCK COMPRESSED

The COLOURING BLOCK COMPRESSED PDU format is shown in Fig. 4.9. Photographic data express the colour and attribute of the colouring blocks in a horizontal line in the FG, BG and display attribute planes.

The first fixed format parameter operand indicates the attribute flags. If the flag of the FG colour, BG colour or display attribute is '1', the PDU contains the corresponding data.

The second parameter, a single-value operand, indicates the absolute coordinate Y where the attribute data decoding is initiated.

Photographic data are composed of attribute-run code pairs.

In an attribute-run pair, four bits are assigned for both attribute and run. A four-bit for FG or BG planes is an entry address of the LUT. Four-bit attributes for display attribute planes are used for flashing controls, and their meaning is shown in Table 4.4. The four-bit run length code table is shown in Table 4.5. The last run in a horizontal line is coded as ECL (End of Colouring Line) regardless of its length.

If the flags of the fixed format operand are all 1s, FG colour-run pairs for the first line come first, BG colour-run pairs for the same line come next and display attribute-run pairs for the same line follow. If there remain more data, they are interpreted as the data for colouring blocks in a line a colouring block height below. The remaining data are interpreted in the same manner as the first line.

If there is flag '0' in the fixed format operand, the data corresponding to it is omitted from the expression.

A coding example is shown in Fig.4.10.

#### 4.7 FIELD COLOURING BLOCK

The PDU format is shown in Fig.4.11. Photographic data express colours and display attributes of the colouring blocks in a field on the FG, BG and display attribute planes.

The first operand (fixed format) indicates the attribute flag in the same manner as the COLOURING BLOCK command. The second and the third are multi-value operands, which specify the

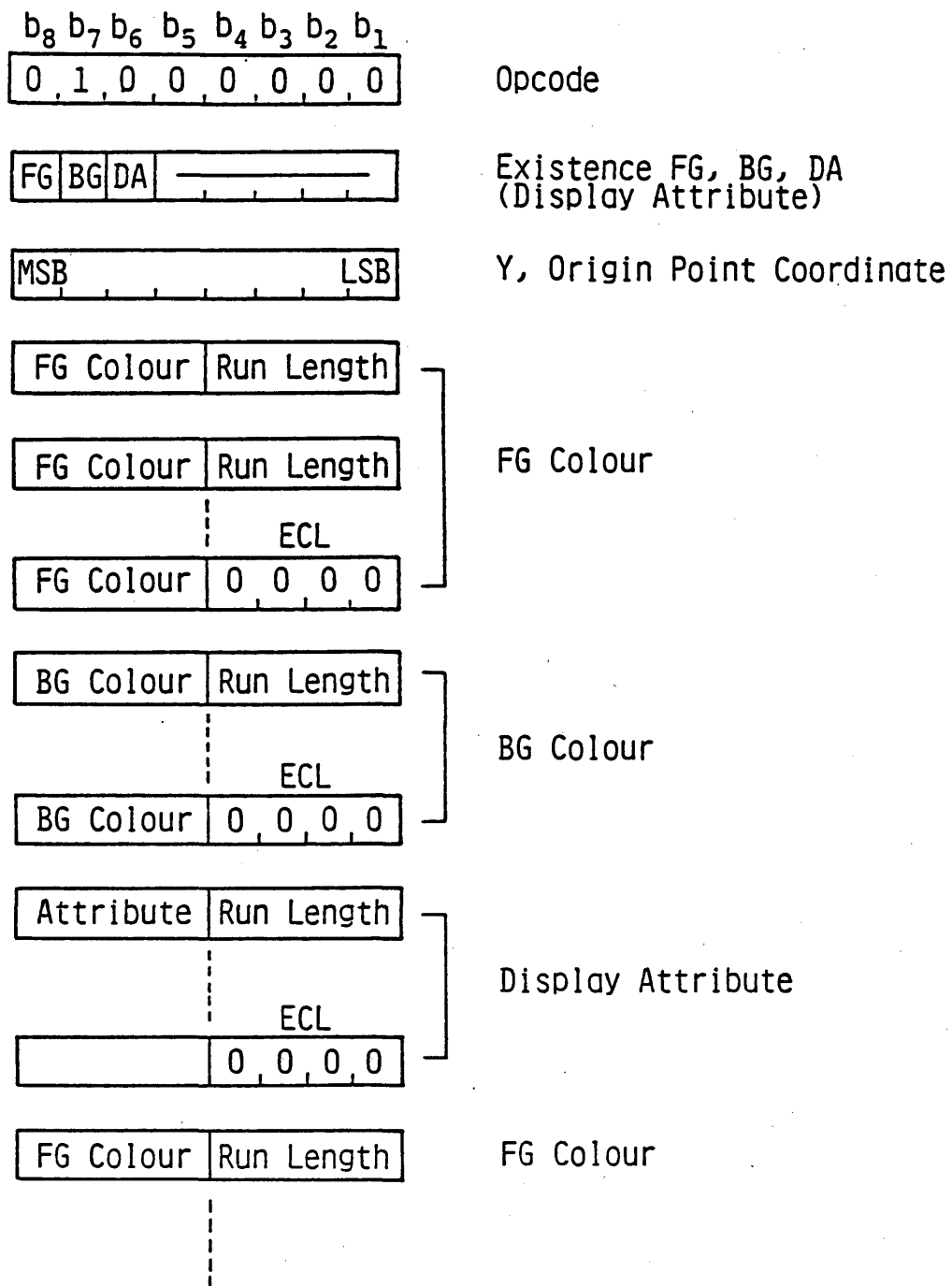
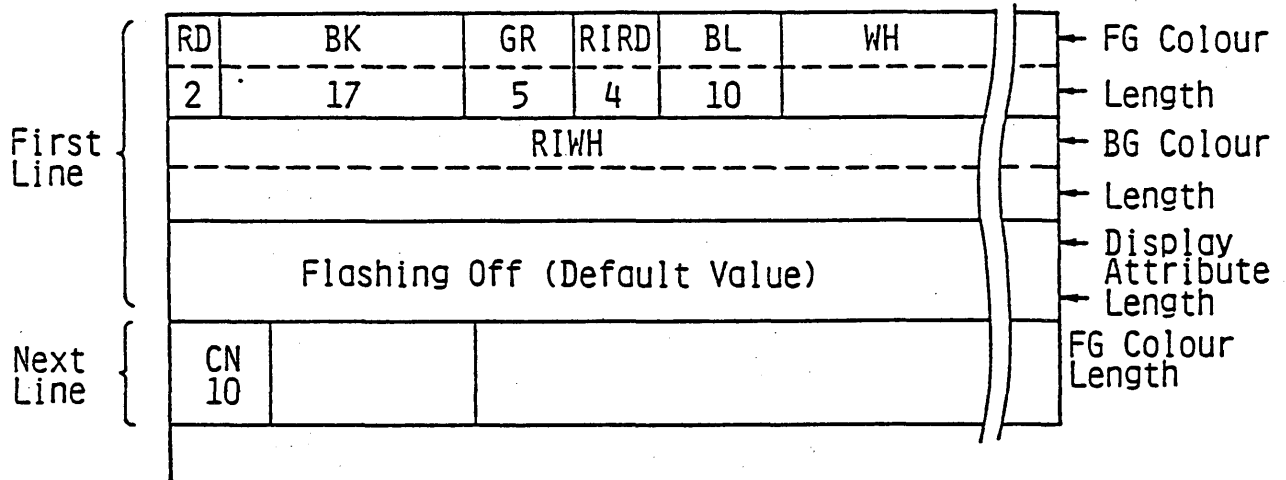


Fig.4.9 COLOURING BLOCK COMPRESSED



Table 4.5 4-bit Run Length Code

Run Length	Code	Run Length	Code
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111
8	1000	ECL	0000



1	1	0							FG/BG	
			RD	0	0	1	0		2	FG Colour
			BK	1	1	1	1		15	
			BK	0	0	1	0		2	
			GR	0	1	1	0		5	
			RIRD	0	1	0	0		4	
			BL	1	0	1	0		10	
			WH	0	0	0	0		ECL	BG Colour
			RIWH	0	0	0	0		ECL	
			CN	1	0	1	0		10	FG Colour

Fig.4.10 A Coding Example of COLOURING BLOCK COMPRESSED

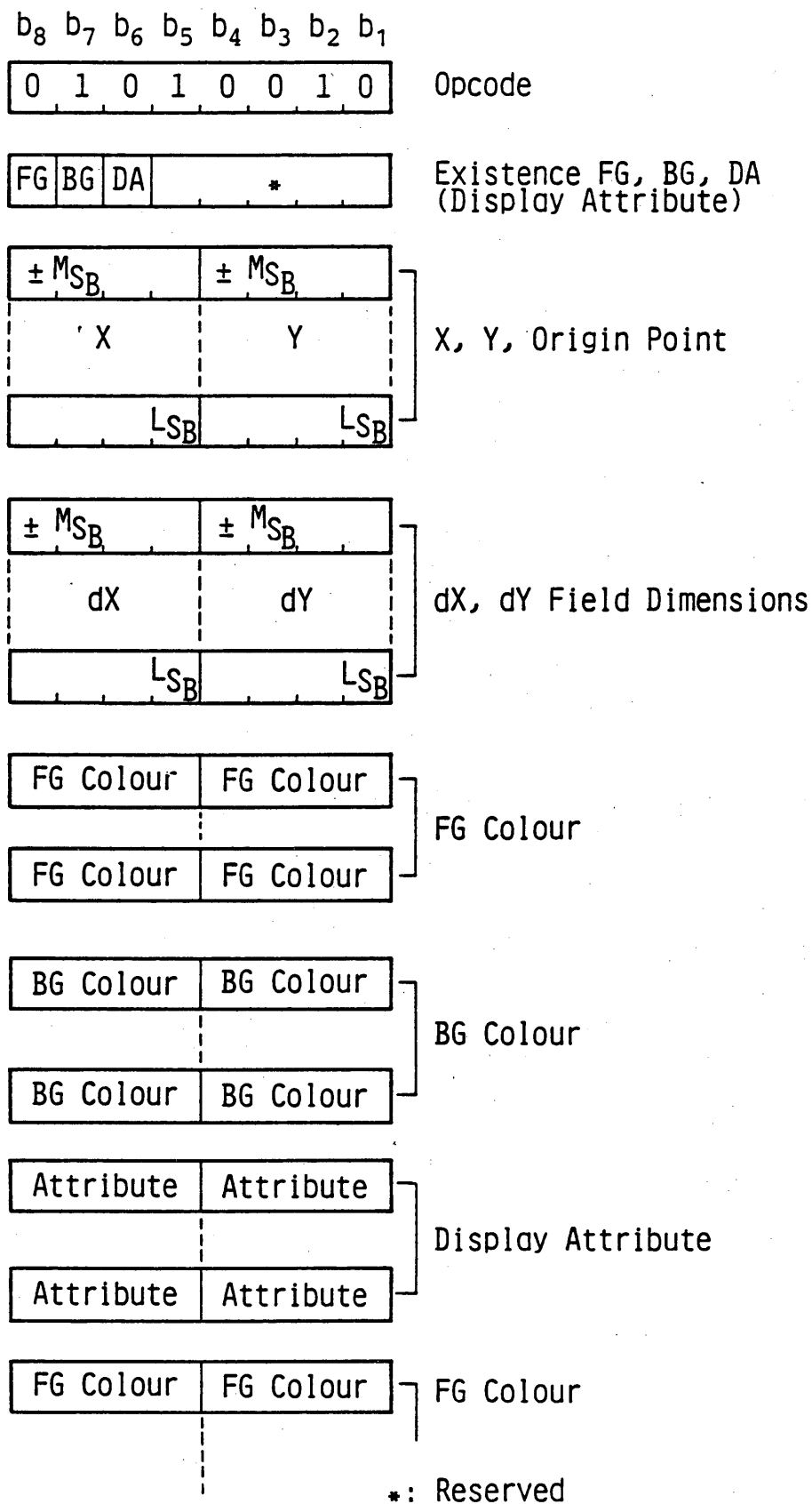


Fig.4.11 FIELD COLOURING BLOCK

location and the dimensions of a rectangular field to which attribute data should be deposited. The initial colouring block position is specified in absolute coordinates by the first multi-value operand, and the width and height (dx, dy) are given by the second multi-value operand. The initial colouring block position may be one of four vertices of the field, since the width and height may be positive or negative.

Photographic data are composed of successive three-four bit codes which represent FG data, BG data and display attributes. A four-bit for FG data or BG data is an entry address of the LUT. Four-bit display attributes are used for flashing controls, and their meaning is shown in Table 4.4.

When the first operand indicates all flags are 1s, FG colour data for the field come first, BG colour data for the same field come next and display attribute data for the same field follow.

When the width (dx) of a field is positive, the drawing position moves to the right next pixel position after data is deposited to a pixel. When dx is negative, the drawing position moves to the left.

When the height (dy) of a field is positive, the drawing position moves to the upper next line, after an entire line within a field is filled. When dy is negative, the drawing position moves to the lower next line. If data overrun the defined field, following data are filled into the field whose parameters are (X + dx, Y) and (dx, dy).

If the field exceeds the block colouring plane boundary, the remaining data are discarded.

Operand data should keep byte boundary in the operand field.

#### 4.8 FIELD COLOURING BLOCK COMPRESSED

The PDU format is shown in Fig.4.12. Photographic data express colours and attributes of the colouring blocks in a field on the FG, BG and display attribute planes.

The first operand (fixed format) indicates the attribute flag in the same manner as the COLOURING BLOCK.

The second and the third are multi-value operands, which specify a rectangular field location and dimensions of the colouring blocks to which attribute data should be deposited. The initial colouring block position is specified in absolute coordinates by the first multi-value operand, and the width and height (dx, dy) are given by the second multi-value operand. The initial colouring block position may be one of four vertices of the field, since the width and height may be positive or negative.

Photographic data following the parameters are composed of attribute-run code pairs, the number of bits of each part is eight (4 bits-4 bits pair), in the same manner as the COLOURING

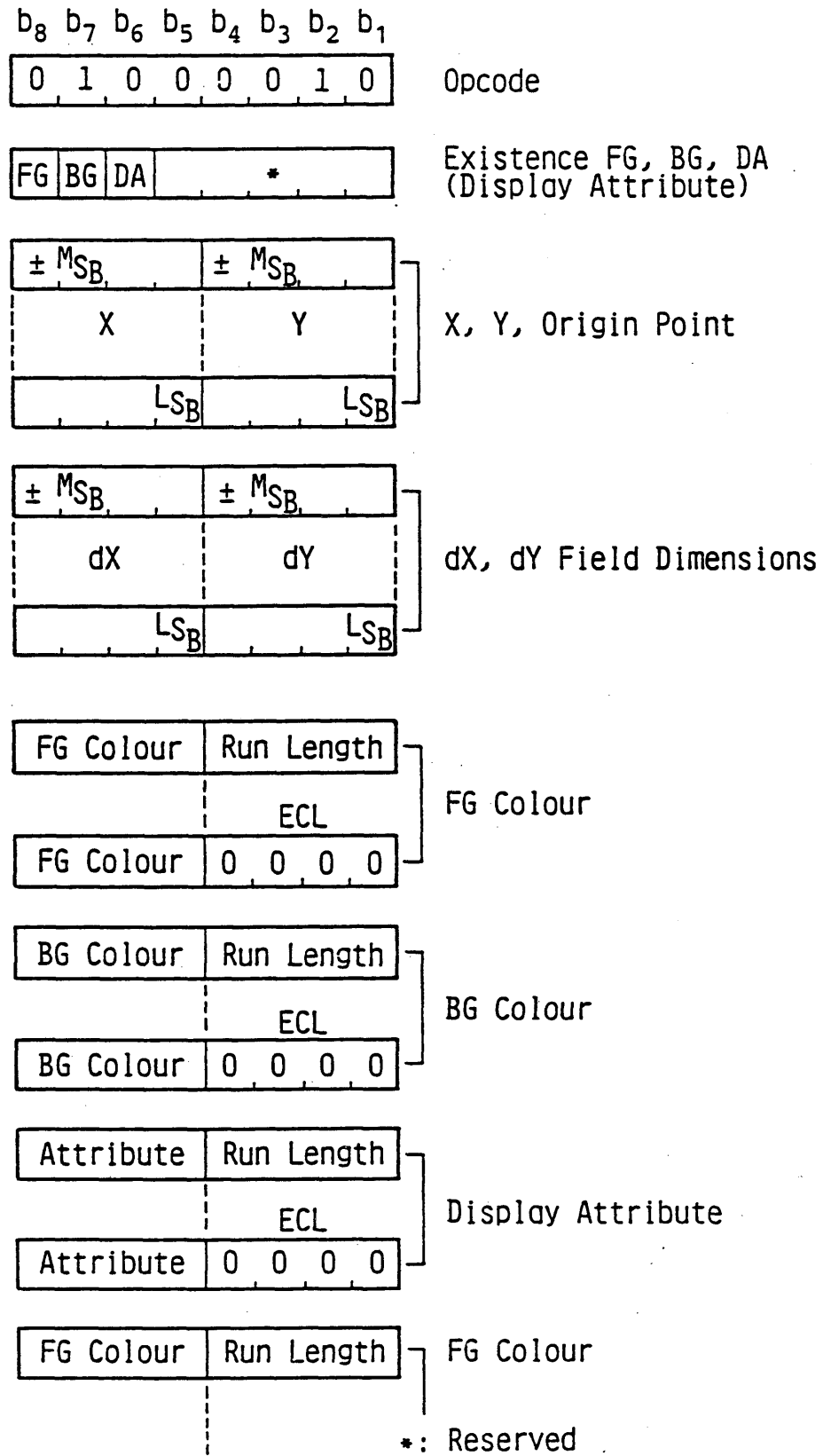


Fig.4.12 FIELD COLOURING BLOCK COMPRESSED

BLOCK COMPRESSED. The last run in the defined field is coded as ECL.

The colouring block position is automatically moved in the X direction a distance equal to the width ( $dx_0$ ) of the colouring block. If the field dimension  $dx$  is positive, position moves to the right and if  $dx$  is negative, position moves to the left.

If the block position reaches or exceeds a vertical side of the field, x coordinate of the colouring block is moved to the other side of the field and the y coordinate is moved by the height ( $dy$ ) of the colouring block. If the field dimension  $dy$  is negative, the block position moves down.

If the colouring block position reaches or exceeds a horizontal side of the field, and if there are remaining photographic data, they should be interpreted as if there proceeds the same opcode with field parameter operands ( $x + dx, y$ ), ( $dx, dy$ ).

#### 4.9 FREE FORMAT COLOURING BLOCK

The PDU format is shown in Fig.4.13. Photographic data is expressed in the same manner as in the FIELD COLOURING BLOCK COMPRESSED, except the bit assignment of an attribute-run code part.

The first fixed format parameter operand indicates the content of photographic data, FG, BG, DA code length and run length expression. The bits and their meanings are shown in Table 4.6.

The run length code specified by the first parameter operand is shown in Table 4.7.

#### 4.10 PHOTO DRCS 1 and PHOTO DRCS 2

These PDU formats are shown in Fig.4.14. Photographic data express the pixel pattern of the character to be defined. The operand is 3 bytes fixed format. The first fixed format parameter operand indicates the number of pixels in a character block for both horizontal and vertical directions. The codes are as follows:

7/3: 16 × 24 pixels (Normal size)

7/8: 8 × 24 pixels (Medium size)

7/1: 8 × 12 pixels (Small size)

The second byte is fixed to 4/1 for DRCS 1, or it represents first byte code for DRCS 2. The third byte is DRCS code for DRCS 1 or second byte code for DRCS 2.

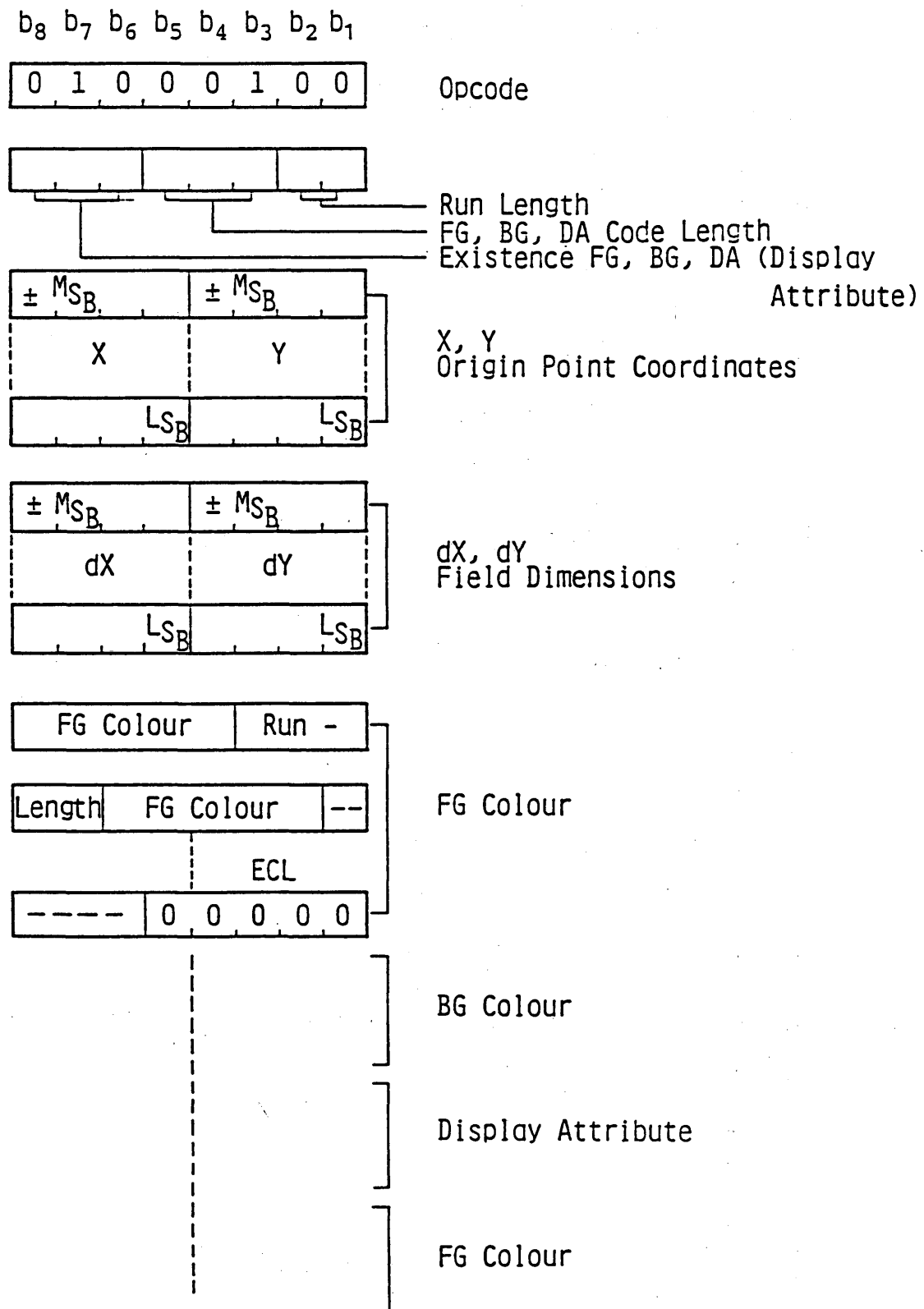


Fig.4.13 FREE FORMAT COLOURING BLOCK

Table 4.6 Fixed Format Byte Interpretation of  
FREE FORMAT COLOURING BLOCK

b8	b7	b6	Data Contents
0	0	1	DA (Display Attributes)
0	1	0	BG
0	1	1	BG and DA
1	0	0	FG
1	0	1	FG and DA
1	1	0	FG and BG
1	1	1	FG, BG and DA
b5	b4	b3	FG/BG/DA Code Length in Bits
0	0	0	3
0	0	1	4
0	1	0	5
0	1	1	6
1	0	0	7
1	0	1	8
1	1	0	9
1	1	1	10
b2	b1		Run Length Code's Length in Bits
0	0		5
0	1		6
1	0		7
1	1		8



Table 4.7 Run Length Code for FREE FORMAT  
COLOURING BLOCK

Code	Run Length
00-----00	ECL
00-----01	1
00-----10	2
00-----11	3
-	-
-	-
-	-

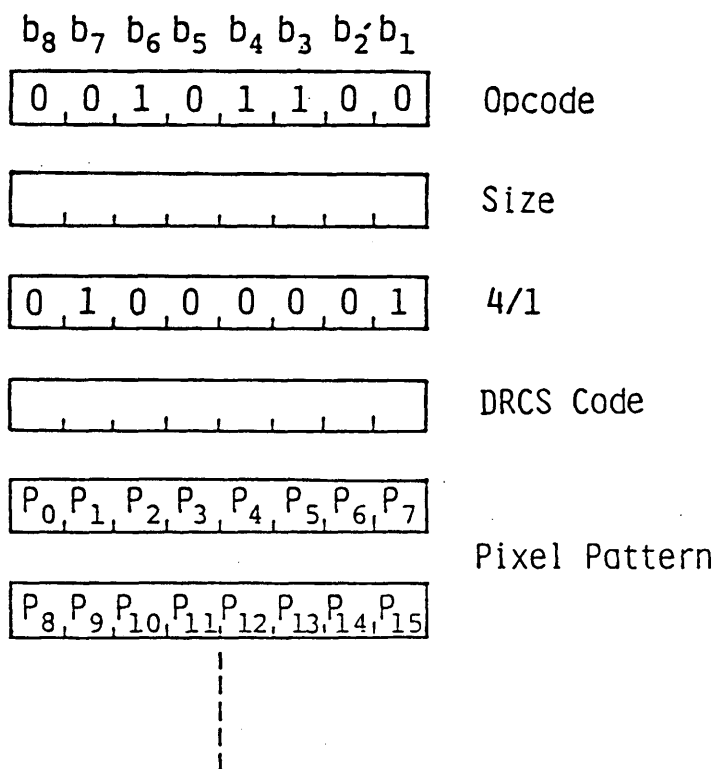


Fig.4.14(1) PHOTO DRCS 1

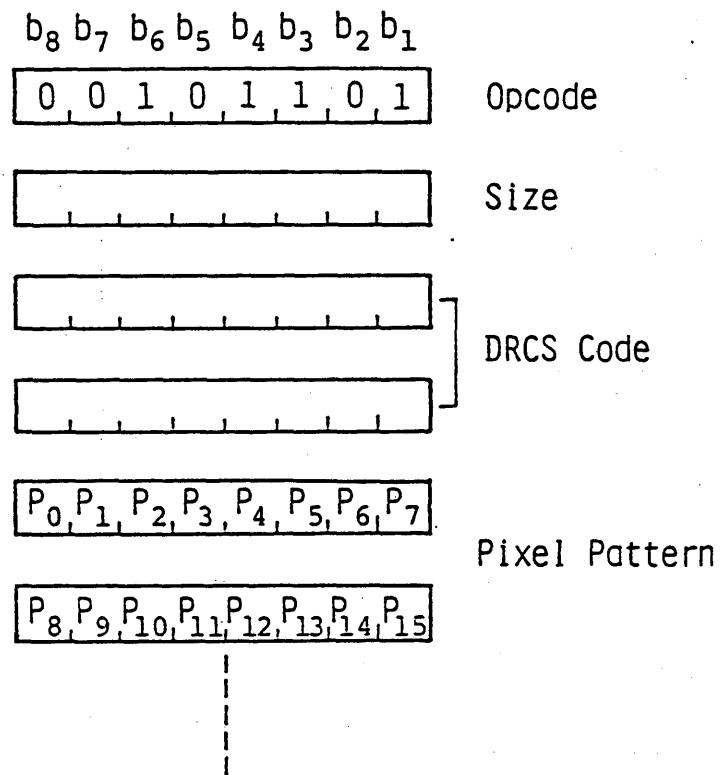


Fig.4.14(2) PHOTO DRCS 2

## 5. Coding in the musical mode

### 5.1 General

Musical note data is conveyed in the musical mode. Although data is character coded in the musical mode, this standard specifies the mode apart from the character code mode, since information contents are different from others. A C-set and a G-set different from the character code mode are set upon invocation of the musical mode.

Fig.5.1 illustrates the table setting in the musical mode.

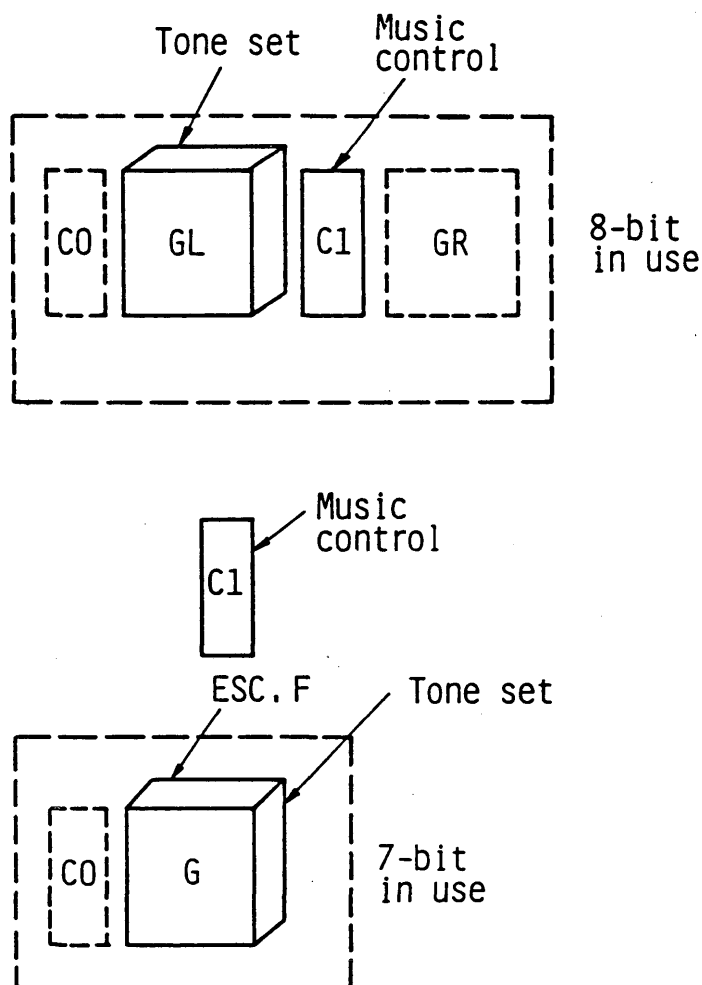


Fig.5.1 Table Setting in the Musical Mode

## 5.2 Musical tone set

The musical tone set is a two-bytes code set which represents pitch and duration of a musical tone. The musical tone set is shown in Table 5.1.

The first byte of a code specifies the tone pitch, defining eighty-eight pitches from A0 up to C8.

The frequency of pitch A4 is determined as 440 Hz: The frequency of each pitch is expressed as

$$f(I) = 440 \times 2^{(I-81)/12} \text{ Hz}$$

where I is the decimal expression of the first byte.

IP (Undefined Pitch) (in case the first byte is 7/15) is used for a rhythm part, which has a fixed pitch.

RST (in case the first byte is 2/0) is used to cause a rest. The second byte determines the duration of a sound whose pitch is specified by the first byte. Sound duration is expressed with a number of Tu: Tu is a unit length of duration. The value of Tu is transmitted prior to tone codes via SMC (See 5.3.1.1.).

## 5.3 Musical control set

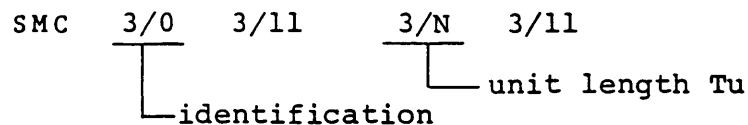
Twelve control codes are used for controlling sound generation. The musical control set is shown in Table 5.2.

### 5.3.1 Start and end of music codes

#### 5.3.1.1 SMC (Start of music codes)

SMC denotes a start of music code sequence. It has two parameters. The first parameter represents coding scheme identification. The following coding scheme is the default: The identification number is 3/0. Other numbers are reserved for future standardization.

The second parameter Tu represents the unit length of sound duration in milli-seconds. (The expression of numeric parameter is described in 5.3.7)



#### 5.3.1.2 EMC (End of music codes)

EMC denotes the end of music codes sequence.

Table 5.1(1) Musical Tone Set (Pitch)

					10	11	12	13	14	15	
					b <sub>7</sub>	0	0	1	1	1	1
					b <sub>6</sub>	1	1	0	0	1	1
					b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7	
0	0	0	0	0	RST	C <sub>2</sub>	E <sub>3</sub>	G <sup>#</sup> <sub>4</sub>	C <sub>6</sub>	E <sub>7</sub>	
0	0	0	1	1	A <sub>0</sub>	C <sup>#</sup> <sub>2</sub>	F <sub>3</sub>	A <sub>4</sub>	C <sup>#</sup> <sub>6</sub>	F <sub>7</sub>	
0	0	1	0	2	A <sup>#</sup> <sub>0</sub>	D <sub>2</sub>	F <sup>#</sup> <sub>3</sub>	A <sup>#</sup> <sub>4</sub>	D <sub>6</sub>	F <sup>#</sup> <sub>7</sub>	
0	0	1	1	3	B <sub>0</sub>	D <sup>#</sup> <sub>2</sub>	G <sub>3</sub>	B <sub>4</sub>	D <sup>#</sup> <sub>6</sub>	G <sub>7</sub>	
0	1	0	0	4	C <sub>1</sub>	E <sub>2</sub>	G <sup>#</sup> <sub>3</sub>	C <sub>5</sub>	E <sub>6</sub>	G <sup>#</sup> <sub>7</sub>	
0	1	0	1	5	C <sup>#</sup> <sub>1</sub>	F <sub>2</sub>	A <sub>3</sub>	C <sup>#</sup> <sub>5</sub>	F <sub>6</sub>	A <sub>7</sub>	
0	1	1	0	6	D <sub>1</sub>	F <sup>#</sup> <sub>2</sub>	A <sup>#</sup> <sub>3</sub>	D <sub>5</sub>	F <sup>#</sup> <sub>6</sub>	A <sup>#</sup> <sub>7</sub>	
0	1	1	1	7	D <sup>#</sup> <sub>1</sub>	G <sub>2</sub>	B <sub>3</sub>	D <sup>#</sup> <sub>5</sub>	G <sub>6</sub>	B <sub>7</sub>	
1	0	0	0	8	E <sub>1</sub>	G <sup>#</sup> <sub>2</sub>	C <sub>4</sub>	E <sub>5</sub>	G <sup>#</sup> <sub>6</sub>	C <sub>8</sub>	
1	0	0	1	9	F <sub>1</sub>	A <sub>2</sub>	C <sup>#</sup> <sub>4</sub>	F <sub>5</sub>	A <sub>6</sub>		
1	0	1	0	10	F <sup>#</sup> <sub>1</sub>	A <sup>#</sup> <sub>2</sub>	D <sub>4</sub>	F <sup>#</sup> <sub>5</sub>	A <sup>#</sup> <sub>6</sub>		
1	0	1	1	11	G <sub>1</sub>	B <sub>2</sub>	D <sup>#</sup> <sub>4</sub>	G <sub>5</sub>	B <sub>6</sub>		
1	1	0	0	12	G <sup>#</sup> <sub>1</sub>	C <sub>3</sub>	E <sub>4</sub>	G <sup>#</sup> <sub>5</sub>	C <sub>7</sub>		
1	1	0	1	13	A <sub>1</sub>	C <sup>#</sup> <sub>3</sub>	F <sub>4</sub>	A <sub>5</sub>	C <sup>#</sup> <sub>7</sub>		
1	1	1	0	14	A <sup>#</sup> <sub>1</sub>	D <sub>3</sub>	F <sup>#</sup> <sub>4</sub>	A <sup>#</sup> <sub>5</sub>	D <sub>7</sub>		
1	1	1	1	15	B <sub>1</sub>	D <sup>#</sup> <sub>3</sub>	G <sub>4</sub>	B <sub>5</sub>	D <sup>#</sup> <sub>7</sub>	IP	

Table 5.1(2) Musical Tone Set (Duration)

					10	11	12	13	14	15	
					b <sub>7</sub>	0	0	1	1	1	1
					b <sub>6</sub>	1	1	0	0	1	1
					b <sub>5</sub>	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		2	3	4	5	6	7	
0	0	0	0	0	Reserved		64Tu	16Tu	32Tu	48Tu	
0	0	0	1	1			1Tu	17Tu	33Tu	49Tu	
0	0	1	0	2			2Tu	18Tu	34Tu	50Tu	
0	0	1	1	3			3Tu	19Tu	35Tu	51Tu	
0	1	0	0	4			4Tu	20Tu	36Tu	52Tu	
0	1	0	1	5			5Tu	21Tu	37Tu	53Tu	
0	1	1	0	6			6Tu	22Tu	38Tu	54Tu	
0	1	1	1	7			7Tu	23Tu	39Tu	55Tu	
1	0	0	0	8			8Tu	24Tu	40Tu	56Tu	
1	0	0	1	9			9Tu	25Tu	41Tu	57Tu	
1	0	1	0	10			10Tu	26Tu	42Tu	58Tu	
1	0	1	1	11			11Tu	27Tu	43Tu	59Tu	
1	1	0	0	12			12Tu	28Tu	44Tu	60Tu	
1	1	0	1	13			13Tu	29Tu	45Tu	61Tu	
1	1	1	0	14			14Tu	30Tu	46Tu	62Tu	
1	1	1	1	15			15Tu	31Tu	47Tu	63Tu	

Tu: Time unit

Note: Columns 2 to 3 are reserved for future extension.

Table 5.2 Musical Control Set

				b <sub>8</sub>	1	1
				b <sub>7</sub>	0	0
				b <sub>6</sub>	0	0
				b <sub>5</sub>	0	1
					8	9
				b <sub>7</sub>	1	1
				b <sub>6</sub>	0	0
				b <sub>5</sub>	0	1
					4	5
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		4/8	5/9
0	0	0	0	0	SMC	SLV
0	0	0	1	1	SMP	
0	0	1	0	2	SRP	
0	0	1	1	3		
0	1	0	0	4	EMC	
0	1	0	1	5	EPT	
0	1	1	0	6		
0	1	1	1	7		
1	0	0	0	8	LBL	LRT
1	0	0	1	9	JMP	
1	0	1	0	10	RPT	
1	0	1	1	11	BRA	
1	1	0	0	12	CTM	
1	1	0	1	13		
1	1	1	0	14		
1	1	1	1	15		

### 5.3.2 Start and end of a part

Each part's information is defined consecutively on a part by part basis. Each definition sequence is separated by the start of part code (SMP or SRP) and the end of part code (EOP).

#### 5.3.2.1 SMP (Start of melody part)

SMP denotes the start of a melody part and specify the timbre played in this melody part. The timbre is defined by the parameter of SMP shown in Table 5.3.

SMP    3/P    3/11  
          |  
          timbre

Table 5.3 Timbre in Melody Part

TIMBRE (INSTRUMENT)	CODE
Violin	3/1 3/0
Guitar	3/1 3/1
(Harpsichord)	
Piano	3/1 3/2
Flute	3/2 3/0
Clarinet	3/2 3/1
Oboe	3/2 3/2
Trumpet	3/2 3/3
Organ	3/3 3/0
Vibraphone	3/4 3/0



#### 5.3.2.2 SRP (Start of rythm part)

SRP denotes the start of a rhythm or percussive part and specify the timbre played in this rhythm part. The timbre is defined by the parameter of SRP shown in Table 5.4.

SRP    3/P    3/11  
          |  
          timbre

Table 5.4    Timbre in Rythm Part

TIMBRE (INSTRUMENT)	CODE
Snare drum	3/1   3/0
Bass drum	3/1   3/1
Tom-tom	3/2   3/0
Suspended cymbal (Top cymbal)	3/3   3/0
High-hat cymbals	3/3   3/1

#### 5.3.2.3 EPT (End of part)

EPT denotes the end of a melody part or a rythnm part.

#### 5.3.3 Repeat and Jump

The following four codes are used to express repeat or jumps. They are followed by one or two numeric parameters, M and/or N.

M indicates a number of repeat or passing time. N does a label number such as a start point of repetition or a destination for a jump.

#### 5.3.3.1 LBL (Label)

Followed by the parameter N, LBL defines a label as the start point of a destination.

LBL    3/N    3/11  
          └label

#### 5.3.3.2 JMP (Jump)

JMP causes jump to the part labeled with the parameter N.

JMP    3/N    3/11  
          └destination (label)

#### 5.3.3.3 RPT (Repeat)

Followed by two parameters M and N, RPT denotes passing after repeating M times. The start point of repetition is indicated by the second parameter N.

RPT    3/M    3/11    3/N    3/11  
          └                  └start point (label)  
                                └number of repeat time

#### 5.3.3.4 BRA (Branch)

Followed by two parameters M and N, BRA denotes a jump to label N just after the Mth passing.

BRA    3/M    3/11    3/N    3/11  
          └                  └destination (label)  
                                └number of passing time

#### 5.3.4 SLV (Sound level)

The sound level (intensity) is denoted by SLV, followed by a parameter shown in Table 5.5. The sound intensity is specified in eight levels, from level 1 to level 8.

Level 1 indicates the minimum intensity and level 8 the maximum.

The sound level differences between adjacent levels is about 3 dB.

SLV    3/L    3/11  
          └level

Table 5.5 Sound Level

LEVEL	CODE
1	3/1
2	3/2
3	3/3
4	3/4
5	3/5
6	3/6
7	3/7
8	3/8

Level 1: lowest  
Level 8: highest  
Level 4: default

#### 5.3.5 CTM (Change of timbre)

The change of timbre in a part is denoted by CTM, followed by a parameter shown in Table 5.3.

CTM    3/P    3/11  
          └ timbre

#### 5.3.6 LRT (Long rest and tone)

LRT is a code for denoting a long tone or a long rest, causing the duration of the next defined tone to be multiplied by N.

LRT    3/N    3/11  
          └ multiplier

### 5.3.7 Numeric parameter

Numeric parameters following a music control code are expressed in binary coded decimals. Characters 3/0 through 3/9 are used to indicate a value and 3/11 is used as a decimal delimiter.

For example, the sequence

<u>3/1, 3/2, 3/3, 3/11,</u>	<u>3/4, 3/5, 3/6, 3/11</u>
$N_1$	$N_2$

is interpreted as

$N_1 = 123$  (Decimal)

and  $N_2 = 456$  (Decimal).

## 6. Default conditions

The default conditions of the table setting, the display attribute, etc. are shown below:

Items	Default condition
Command mode	Character code mode
Graphic Sets	
G0	Japanese-Kanji set
G1	Primary set
G2	Display control command set
G3	DRCS 1
In-use table	
GL	G0
GR	G2
(In 7 bit environment	G0)
Single-value operand	1 byte
Length	
Multi-value operand	3 bytes (Character code mode)
	2 bytes (Transparent mode)
Logical pel sizes	(1/256, -1/256)
Colouring block sizes	(4/256, -4/256)
Character block sizes	(16/256, 24/256)
Foreground colour	White (7th colour)
Background colour	Transparent (8th colour)
LUT contents	As specified in Table 3.5
Raster colour	Reduced intensity blue
Header Raster colour	Reduced intensity blue
Cursor display	Off
Flashing	Off
Conceal	Off
Lining	Off
Protect	Off
Scroll	Off
Blinking	Off
Character size	Normal
Text rotation	0 deg.
Character path	To the right
Inter-character spacing	1
Inter-row spacing	1
Move parameter	Move independently
Cursor Style	Underscore
Cursor position	Home position
Drawing point	(0, 0)
Line texture	Solid
Fill pattern	Solid
Highlight	Off
Active area	The defined display area

Frame index	10
Overwriting mode	Replace
Macros	Null
DRCS 1 and 2	Null

## Appendix 1 Service Reference Model (SRM)

The Service Reference Model defines the recommended features that should be implemented in an ordinary terminal. These sets of requirements also represent the maximum functionalities that the information provider should assume when encoding text and pictorial information. The SRM is intended to give measure for the implementation of a terminal or for page creation, however, it is by no means obligatory.

Functions	Requirement
1. Code extension sequence	All code extension sequences defined in this standard should be executed.
2. C0 control set	The set should be executed as described in this standard.
3. C1 control set	The set should be executed as described in this standard, except that flashing process may be limited to two phase operation. The ON interval and the OFF interval are about 0.5 second each.
4. The graphic character repertoire	All characters of the primary set, the Japanese-Kanji set and the Katakana set should be presented.
5. Mosaic sets	All mosaics of the Mosaic 1 set and the Mosaic 2 set should be presented.
6. DRCSs	Both the DRCS 1 set and the DRCS 2 set should be definable. The number of simultaneously definable characters is subject to memory limitation.
7. Macro set	The number of macros definable should be 96, subject to memory limitation.
8. Memory requirements	The terminal should provide minimum 2 K bytes memory shared for both DRCS and Macros.

9. Display control command set

Following commands should be implemented as specified in this standard.

P-RESET  
P-DOMAIN  
LOGICAL PEL  
DISPLAY MODE  
RASTER  
HEADER RASTER

10. Photographic commands

Following commands should be implemented as specified in this standard.

LINE DOT PATTERN  
LINE DOT PATTERN COMPRESSED  
FIELD DOT PATTERN  
COLOURING BLOCK  
COLOURING BLOCK COMPRESSED  
FIELD COLOURING BLOCK  
PHOTO DRCS 1  
PHOTO DRCS 2

11. Physical display parameters

Resolution should be 248 pixels horizontal by 204 pixels vertical including 'Header area'.



## Appendix 2    Operation of CS and NSR

Both CS and NSR reestablish following attributes.

- Graphic set table setting
- Domain
- Logical pel sizes
- Colouring block sizes
- Foreground colour
- Background colour
- Cursor display
- Flashing
- Conceal
- Lining
- Protect
- Scroll
- Character block sizes
- Text display attributes
- Character size
- Drawing point
- Line texture
- Fill pattern
- Highlight
- Frame index
- Overwriting mode

Following attributes are set to default states only by CS.

- Raster colour
- Header raster colour
- Cursor position (NSR sets the cursor position according to its parameters)
- LUT contents
- Blink process
- Active area
- Raster memory contents
- DRCS definition
- MACRO definition

