



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.



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

# CCITT

COMITÉ CONSULTATIF  
INTERNATIONAL  
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

ANNEXE AU LIVRE BLEU

---

FASCICULE VII.5 — ANNEXE I



## SYNTAXE DE DONNÉES I POUR LE SERVICE VIDÉOTEX INTERACTIF INTERNATIONAL

RECOMMANDATION T.101, ANNEXE B

---



IX<sup>e</sup> ASSEMBLÉE PLÉNIÈRE  
MELBOURNE 1988

Genève 1990



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

# CCITT

COMITÉ CONSULTATIF  
INTERNATIONAL  
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

**ANNEXE AU LIVRE BLEU**

---

**FASCICULE VII.5 — ANNEXE I**



**SYNTAXE DE DONNÉES I  
POUR LE SERVICE VIDÉOTEX  
INTERACTIF INTERNATIONAL**

**RECOMMANDATION T.101, ANNEXE B**

---



**IX<sup>e</sup> ASSEMBLÉE PLÉNIÈRE  
MELBOURNE 1988**

Genève 1990



## ANNEXE B

(de la Recommandation T.101)

### SYNTAXE DE DONNÉES I

*Remarque* – Cette syntaxe de données correspond généralement à la syntaxe de données de couche présentation «CAPTAIN», officiellement adoptée par le JAPON.

#### *Préface*

La présente annexe décrit les formats, règles et procédures pour le codage d'information sous forme de texte, d'image ou de musique pour application vidéotexte. Cette spécification est conforme à l'architecture définie dans les modèles de référence à plusieurs couches d'interconnexion de systèmes ouverts de l'ISO et du CCITT comme faisant partie d'un protocole de niveau de présentation.

L'exploitation est possible dans un environnement à 7 bits ou 8 bits. Pour les informations sous forme de texte, des caractères japonais, des alphanumériques et des jeux de caractères dynamiquement redéfinissables (JCDR pour un multiplète comme pour deux multiplètes) sont prévus. Pour les informations sous forme d'image, le schéma de codage photographique, qui est une caractéristique spéciale de cette syntaxe de données, est fourni. Le codage de mosaïque et le codage géométrique sont également disponibles pour les informations sous forme d'image. Il existe deux ensembles de mosaïque, dont l'un est compatible avec la syntaxe de données II. Le schéma de codage géométrique est conforme à la syntaxe de données III. Les possibilités supplémentaires comprennent table de consultation de couleur, macros, clignotement, champs non protégés, caractères dissimulés, défilement et taille de caractère variable. Le codage de notes musicales et le codage d'images animées sont disponibles en option.

## SOMMAIRE

	<i>Page</i>	
1	Généralités	6
1.1	Portée	6
1.2	Références	6
1.3	Définitions	6
1.4	Affichage d'informations visuelles	8
1.4.1	Structure d'affichage	8
1.4.2	Structure de trame logique	8
1.4.2.1	Trame de coloration par blocs	8
1.4.2.2	Trame de coloration par points	9
1.4.3	Système de coordonnées	9
1.5	Information musicale	9
2	Architecture de codage	9
2.1	Généralités	9
2.2	Technique de commutation de mode	9
2.3	Structure de codage dans le mode code de caractère	12
2.4	Structure de codage dans le mode transparent	16
2.5	Structure de codage dans le mode musical	16
3	Codage dans le mode code de caractère	17
3.1	Ensemble de commandes C0	17
3.2	Ensemble de commandes C1	19
3.2.1	Commandes de couleur	20
3.2.2	Commandes de taille de caractère	21
3.2.3	Commandes de curseur	22
3.2.4	Commandes de clignotement	22
3.2.5	Commandes de répétition	22
3.2.6	Commandes de traçage de ligne	22
3.2.7	Commandes de dissimulation	22
3.2.8	Commandes de protection	23
3.2.9	Instruction macro photo	23
2	<b>Fascicule VII.5 – Annexe B</b>	

	<i>Page</i>
3.2.9.1 Généralités	23
3.2.9.2 DEF MACRO P	23
3.2.9.3 DEFP MACRO P	23
3.2.9.4 DEFT MACRO P	23
3.2.9.5 FIN DE MACRO P	24
3.3 Ensemble d'instructions de commandes d'affichage	24
3.3.1 Généralités	24
3.3.2 Fonctions de commande d'attributs	27
3.3.2.1 DOMAINE P	27
3.3.2.2 PEL LOGIQUE	29
3.3.2.3 Commandes du tableau de consultation des couleurs (LUT)	29
3.3.2.3.1 INITIALISATION DE LUT	29
3.3.2.3.2 CLIGNOTEMENT P	29
3.3.2.4 Commandes de couleur de trame	31
3.3.2.5 RÉINITIALISATION P	31
3.3.2.6 MODE D'AFFICHAGE	34
3.3.2.7 TEXTE P	35
3.3.2.8 ATTENTE P	38
3.3.3 Fonctions de commande multitrane	39
3.3.3.1 Généralités	39
3.3.3.2 Flux général des opérations d'affichage multitrane	39
3.3.3.3 ZONE	39
3.3.3.4 INITIALISATION DE TRAME	41
3.3.3.5 ASSIGNATION DE TRAME	42
3.4 Jeux de caractères	43
3.4.1 Jeu de caractères Kanji-japonais	43
3.4.2 Jeu de caractères primaires	43
3.4.3 Jeu de caractères Katakana	43

3.5	Jeu de mosaïques	43
3.5.1	Jeu de mosaïques 1.	43
3.5.2	Jeu de mosaïques 2	43
3.6	Jeux de caractères dynamiquement redéfinissables (JCDR)	51
3.6.1	JCDR 1	51
3.6.2	JCDR 2	51
3.7	Jeu de macros	51
3.8	Jeu d'instructions de description d'image (PDI)	51
3.9	Jeu d'instructions de mouvement (MVI)	51
3.9.1	DÉBUT MVI	54
3.9.2	FIN MVI	54
3.9.3	DÉBUT RÉPÉTITION	56
3.9.4	FIN RÉPÉTITION	56
3.9.5	MOUVEMENT	56
4	Codage en mode transparent	57
4.1	Unité de données photographiques (PDU)	57
4.2	CONFIGURATION DE POINTS EN LIGNE	58
4.3	CONFIGURATION DE POINTS EN LIGNE AVEC COMPRESSION	58
4.4	CONFIGURATION DE POINTS EN CHAMP	61
4.5	BLOC DE COLORATION	65
4.6	BLOC DE COLORATION AVEC COMPRESSION	66
4.7	BLOC DE COLORATION DE CHAMP	67
4.8	BLOC DE COLORATION DE CHAMP AVEC COMPRESSION	71
4.9	BLOC DE COLORATION DE FORMAT LIBRE	73
4.10	JCDR 1 PHOTO ET JCDR 2 PHOTO	73
4.11	Télélogiciel	78
5	Codage dans le mode musical	79
5.1	Généralités	79
5.2	Jeu de tonalités musicales	79

	<i>Page</i>	
5.3	Jeu de commandes musicales	82
5.3.1	Départ et fin de codes de musique	82
5.3.1.1	SMC (Départ des codes de musique)	82
5.3.1.2	EMC (Fin des codes de musique)	82
5.3.2	Départ et fin d'une partie	82
5.3.2.1	SMP (Départ de partie de mélodie)	82
5.3.2.2	SRP (Départ de partie de rythme)	84
5.3.2.3	EPT (Fin de partie)	85
5.3.3	Répétition et saut	85
5.3.3.1	LBL (Etiquette)	85
5.3.3.2	JMP (Saut)	85
5.3.3.3	RPT (Répétition)	85
5.3.3.4	BRA (Branchement)	85
5.3.4	SLV (Niveau sonore)	85
5.3.5	CTM (Changement de timbre)	86
5.3.6	LRT (Longue pause et tonalité)	86
5.3.7	Paramètre numérique	86
6	Conditions de défaut	87
	<i>Appendice 1 – Modèle de référence de service (SRM)</i>	88
	<i>Appendice 2 – Fonctionnement de CS et de NSR</i>	89

## 1 Généralités

### 1.1 Portée

La présente norme décrit la syntaxe de données I, annexe B de la Recommandation T.101, qui permet l'échange d'informations sous forme de texte, d'image et de son dans le réseau de service vidéotex international.

### 1.2 Références

La présente norme est censée être aussi compatible que possible avec les Recommandations et norme suivantes:

Recommandation T.50 (CCITT*)	Alphabet international n° 5
Recommandation F.300 (CCITT)	Service vidéotex
Recommandation S.100 (CCITT)	Echange international d'informations pour vidéotex interactif
Norme ISO** 2022 – 1982	Jeux ISO de caractères codés à 7 et à 8 éléments – Techniques d'extension de code
Norme ISO 6937 – 1982	Traitement de l'information – Jeux de caractères codés pour la transmission de texte

### 1.3 Définitions

Les définitions suivantes sont utilisées dans la présente norme:

**ATTRIBUT:** paramètre réglable appliqué aux informations sous forme de texte, d'image ou de son.

**COMBINAISON DE BITS:** ensemble ordonné de bits (chiffres binaires) qui représente un caractère ou une fonction de commande.

**ENSEMBLE-C:** «ensemble de commandes»; il y a deux ensembles de commandes, C0 et C1, chacun comprenant 32 positions de caractère disposées en 2 colonnes et 16 rangées.

**BLOC DE CARACTÈRE:** zone rectangulaire à l'intérieur de laquelle est affiché un caractère de taille normale.

**MODE DE CODE DE CARACTÈRE:** mode dans lequel la structure de codage se base sur la technique d'extension de code de la norme ISO 2022.

**INSTRUCTION DE MODE DE CODE DE CARACTÈRE:** caractère ou instruction, par exemple, l'une des instructions de commande de visualisation, codé dans le mode de code de caractère.

**CHAMP DE CARACTÈRE:** zone rectangulaire à l'intérieur de laquelle un caractère est affiché à la taille de caractère couramment défini.

**EXTENSION DE CODE:** techniques permettant de développer l'espace d'adresse de caractère absolu d'un code d'octets dans un espace d'adresse virtuelle plus grand.

**TABLEAU DE CODE:** ensemble de règles sans ambiguïté qui définit le mappage entre les combinaisons de bit reçues et les caractères au niveau présentation.

**ZONE D'AFFICHAGE DÉFINIE:** zone adressable de l'écran d'affichage physique, dans laquelle l'écran de l'unité ou une portion de l'écran de l'unité est mappé. La zone d'en-tête n'est pas comprise dans cette surface.

**DÉSIGNER:** identifier un jeu donné dans le répertoire des jeux-G: G0, G1, G2 ou G3.

**JEU D'INSTRUCTIONS DE COMMANDE DE VISUALISATION:** jeu-G. Une instruction de commande se compose d'un code d'opération suivi de zéro ou plusieurs opérandes et définit une fonction de commande d'attribut ou une structure multiframe.

---

\* Comité consultatif international télégraphique et téléphonique.

\*\* Organisation internationale de normalisation.

**JEU DE CARACTÈRES DYNAMIQUEMENT REDÉFINISSABLES (JCDR):** jeu-G comprenant des caractères définissables dont les modèles peuvent être chargés à partir de l'ordinateur principal (hôte).

**SÉQUENCE D'ÉCHAPPEMENT:** chaîne de deux combinaisons de bits ou plus commençant par le caractère ESC. Les formats et les règles concernant l'utilisation des séquences d'échappement sont spécifiés dans la norme ISO 2022.

**TRAME:** unité minimum de la structure de visualisation, sur laquelle l'information complète, mais pas nécessairement toute l'information, peut être affichée. Toute l'information sur l'écran peut être composée de quelques trames. Une utilisation typique d'une trame nécessite plusieurs plans de mémoire.

**JEU-G:** un des quatre ensembles, G0, G1, G2 et G3, chacun comprenant 94 ou 96 positions de caractère disposées en 6 colonnes et 16 rangées.

**ZONE D'EN-TÊTE:** zone utilisée pour la visualisation des messages du système et le moniteur d'entrée par clavier.

**JEU DE CARACTÈRES KANJI-JAPONAIS:** jeu-G à 2 multiplats. Le jeu complet, soit 3657 caractères au total comprenant les caractères kanji-japonais, les signes phonétiques japonais (Katakana et Hirakana), les caractères romains, les nombres et des caractères supplémentaires, est défini.

**JEU DE CARACTÈRES KATAKANA:** jeu-G qui définit 63 signes phonétiques japonais.

**COUCHE:** terme adopté par l'ISO pour décrire chaque module individuel du modèle de référence pour l'interconnexion des systèmes ouverts (OSI) (on emploie indifféremment les termes «niveau» et «couche»).

**FONCTION DE GLISSEMENT:** invocation d'un ensemble de code dans la table utilisée qui reste en vigueur jusqu'à ce qu'un autre ensemble de code soit invoqué à sa place.

**MOSAÏQUE:** matrice rectangulaire d'éléments prédéfinis pouvant être utilisés pour la construction d'images graphiques en bloc.

**INSTRUCTION DE MOUVEMENT (MVI):** code d'opération suivi par zéro ou plusieurs opérandes qui constitue un mouvement de trame exécutable ou une instruction de commande.

**ENSEMBLE DE COMMANDES MUSICALES:** ensemble-C. Les codes de commande sont utilisés pour commander la production de sons musicaux.

**MODE MUSICAL:** mode dans lequel l'information musicale est transportée. Les notes musicales sont codées avec les techniques de codage de caractère.

**ENSEMBLE DE TONALITÉS musicales:** ensemble-G à deux multiplats qui définit la hauteur et la durée d'un son musical.

**UNITÉ DE DONNÉES PHOTOGRAPHIQUES (PDU):** code d'opération avec indicateur de longueur (LI), suivi par zéro, un ou plusieurs opérandes, chacun se composant d'un ou plusieurs multiplats de séquences de bits. Cette utilisation de tous les schémas d'octet possibles est permise dans les séquences de bits d'opérande, ce qui se traduit par une expression efficace de données arbitraires. Le code d'opération se compose d'un octet unique qui indique le sens des données photographiques contenues dans la PDU. Le LI comprend un ou plusieurs octets. La valeur d'un LI est un chiffre binaire qui représente la longueur totale des opérandes suivant le champ LI dans l'octet. Un ou plusieurs octets de paramètres se situent dans la partie d'attaque du champ d'opérande. Les paramètres comprennent les coordonnées de point de dessin où les données photographiques doivent être affichées, et/ou le format de disposition qui indique la manière dont les données photographiques sont disposées. Les données photographiques exprimées sur la base point par point sont contenues dans la partie restante du champ d'opérande.

**INSTRUCTION DE DESCRIPTION D'IMAGE (PDI):** code d'opération suivi par zéro ou plusieurs opérandes, qui constitue un dessin d'image exécutable ou une instruction de commande.

**PLAN (ou PLAN DE MÉMOIRE):** partie de trame, contenant un composant d'information tel qu'une information de modèle, une information de couleur de premier plan, etc.

**NIVEAU DE PRÉSENTATION (ou COUCHE):** sixième des sept niveaux de protocole définis par le modèle de référence ISO d'interconnexion des systèmes ouverts. Le niveau de présentation est essentiellement responsable du codage de l'information textuelle, graphique et de commande d'affichage.

**JEU DE CARACTÈRES PRIMAIRES:** jeu-G qui définit 52 caractères romains, 10 chiffres et 32 marques.

**PROTOCOLE:** ensemble de formats, de règles et de procédures régissant l'échange d'information entre des processus homologues au même niveau.

**MODÈLE DE RÉFÉRENCE DE SERVICE (SRM):** fonctions recommandées devant être mises en oeuvre par un terminal ou un décodeur ordinaire.

**GLISSEMENT UNIQUE:** invocation d'un ensemble de code à l'intérieur de la table utilisée qui n'affecte que l'interprétation de la prochaine combinaison de bit reçue. L'interprétation retourne automatiquement au contenu précédent de la table (on parle également de glissement non bloquant).

**ÉQUIPEMENT TERMINAL:** équipement permettant l'échange de combinaisons de bits codées par télécommunication ou par échange physique de supports de mémoire.

**MODE TRANSPARENT:** mode dans lequel tous les bits du niveau présentation sont utilisés pour des données d'image, de son ou télélogiciel. Ce mode procure un moyen efficace pour la transmission de quantité relativement importante de données.

**COMMANDE DE MODE TRANSPARENT:** commande codée en mode transparent.

**ÉCRAN UNITAIRE:** espace d'adresse d'affichage logique à l'intérieur duquel toutes les opérations de dessin sont exécutées et les caractères alphanumériques déposés. Les dimensions de l'écran unitaire sont 0 à 1 dans les dimensions horizontale, verticale (Y) et de profondeur (Z). (La dernière n'est définie que dans le mode à trois dimensions.)

**ZONE D'AFFICHAGE VISIBLE:** totalité de l'écran d'affichage physique visible pour l'utilisateur.

## 1.4 *Affichage d'informations visuelles*

### 1.4.1 *Structure d'affichage*

La présente norme permet une structure multitrame (voir la figure 1.1) qui procure une superposition de plusieurs trames.

Chaque trame peut être de construction différente: une trame à coloration par points possède des informations de couleur point par point, alors qu'une trame à coloration par blocs en possède bloc par bloc.

Tout le niveau d'arrière-plan de l'écran possède la priorité d'affichage la plus basse: la couleur de la trame et la couleur de la trame d'en-tête ne sont affichées qu'aux emplacements où la trame supérieure est «transparente» et en dehors de la zone d'affichage définie.

La manipulation multitrame est obtenue au moyen de certaines instructions de commande d'affichage. Les détails sont donnés au § 3.3.3.

### 1.4.2 *Structure de trame logique*

#### 1.4.2.1 *Trame de coloration par blocs*

Une trame de coloration par blocs se compose d'un plan modèle, d'un plan de couleur de premier plan (FG), d'un plan de couleur d'arrière-plan (BG) et d'un plan d'attribut d'affichage (DA) comme illustré par la figure 1.1.

L'information relative aux couleurs dans le plan FG et le plan BG est donnée par blocs de coloration. Le bloc de coloration est une zone rectangulaire, normalement plus grande que l'élément d'image de plan de modèle et est une unité minimum de définition de couleur.

Le plan de modèle est un plan de mémoire d'une profondeur de un bit. Lorsque le bit est «1», la couleur du premier plan est affichée. Lorsque le bit est «0», c'est la couleur de l'arrière-plan qui est affichée.

Un plan d'attribut d'affichage possède des drapeaux pour les attributs de clignotement, de dissimulation et de protection par blocs de coloration.<sup>1)</sup>

---

<sup>1)</sup> Clignotement: les schémas sont affichés alternativement dans la couleur du premier plan et dans celle de l'arrière-plan.

Dissimulation: les schémas sont rendus invisibles jusqu'à ce que l'attribut de dissimulation soit annulé.

Protection: un bloc de caractères ou un bloc de coloration est protégé contre toute altération par l'utilisateur.

La structure de trame de coloration par blocs permet une économie de la durée de transmission de données et de la quantité de mémoire de trame tout en maintenant la qualité de la définition (résolution) des couleurs. Le texte et les images sont normalement dessinés sur ce type de trame.

#### 1.4.2.2 *Trame de coloration par points*

Une trame de coloration par points permet d'enregistrer en mémoire des informations de couleur sur la base d'éléments d'image. Le meilleur moyen de reproduire des images codées géométriquement et de le faire sur une trame de coloration par points (voir la figure 1.1).

#### 1.4.3 *Système de coordonnées*

Le système de coordonnées utilisé dans les deux types de données d'image photographique et géométrique se base sur le concept de l'écran unitaire. Un pel logique ou une primitive géométrique se trouve à tout moment à l'intérieur de l'écran unitaire. Ses coordonnées horizontale (X) et verticale (Y) varient de 0 à 1. L'origine (0, 0) de l'écran unitaire se trouve dans le coin inférieur gauche de l'écran d'affichage physique.

L'écran d'affichage physique doit couvrir au moins la zone rectangulaire  $0 \leq X \leq 0,969$ ,  $0 \leq Y \leq 0,797$ . L'information normale est affichée dans la région  $0 \leq y \leq 0,75$ , alors que le reste de l'écran ( $0,75 \leq y \leq 0,797$ : en-tête) est utilisé pour l'affichage des messages du système et pour le moniteur d'entrée à clavier.

Le niveau d'arrière-plan plein écran est divisé en deux régions: la trame ( $y \leq 0,75$ ) et la trame d'en-tête ( $0,75 \leq y$ ).

Le concept d'écran unitaire et de niveau d'arrière-plan plein écran est décrit sur les figures 1.2 et 1.3.

#### 1.5 *Information musicale*

L'information musicale est codée en codes de pseudo-caractères et est chargée dans un terminal. Ensuite, un générateur de tonalité multicanal installé dans le terminal peut jouer la musique.

La technologie de commutation de mode est employée pour présenter l'information musicale (voir le § 2.2).

## 2 **Architecture de codage**

### 2.1 *Généralités*

L'information sous forme de texte est codée en caractères et les primitives géométriques, les notes musicales ainsi que les instructions de commande d'affichage sont codées selon le même principe que les codes de caractère. Les deux environnements 7 bits et 8 bits sont utilisables.

Par ailleurs, les données d'image photographique sont codées selon un principe de codage transparent de façon à enregistrer en mémoire et à transmettre de manière efficace des quantités de données relativement importantes.

### 2.2 *Technique de commutation de mode*

La présente norme spécifie trois modes de fonctionnement distincts: le mode code de caractère, le mode transparent et le mode musical. Le mode code de caractère est le mode dans lequel la structure de codage se base sur la technique d'extension de code de la norme ISO 2022.

Dans le mode transparent, il est possible de transmettre des données transparentes.

Dans le mode musical, les notes musicales sont codées et transmises avec la technique de codage de caractère; cependant, l'ensemble-C et l'ensemble-G sont différents de ceux utilisés dans le mode de code de caractère.

On appelle chacun de ces modes à l'aide de l'identificateur de données (DI) et de l'indicateur de longueur (LI).

L'indicateur de longueur (LI) spécifie le nombre d'octets dans un mode. Si l'expression décimale du LI est 255, les deux octets suivants (le troisième et le quatrième) indiquent la longueur réelle.

La figure 2.1 illustre le concept de commutation de mode.

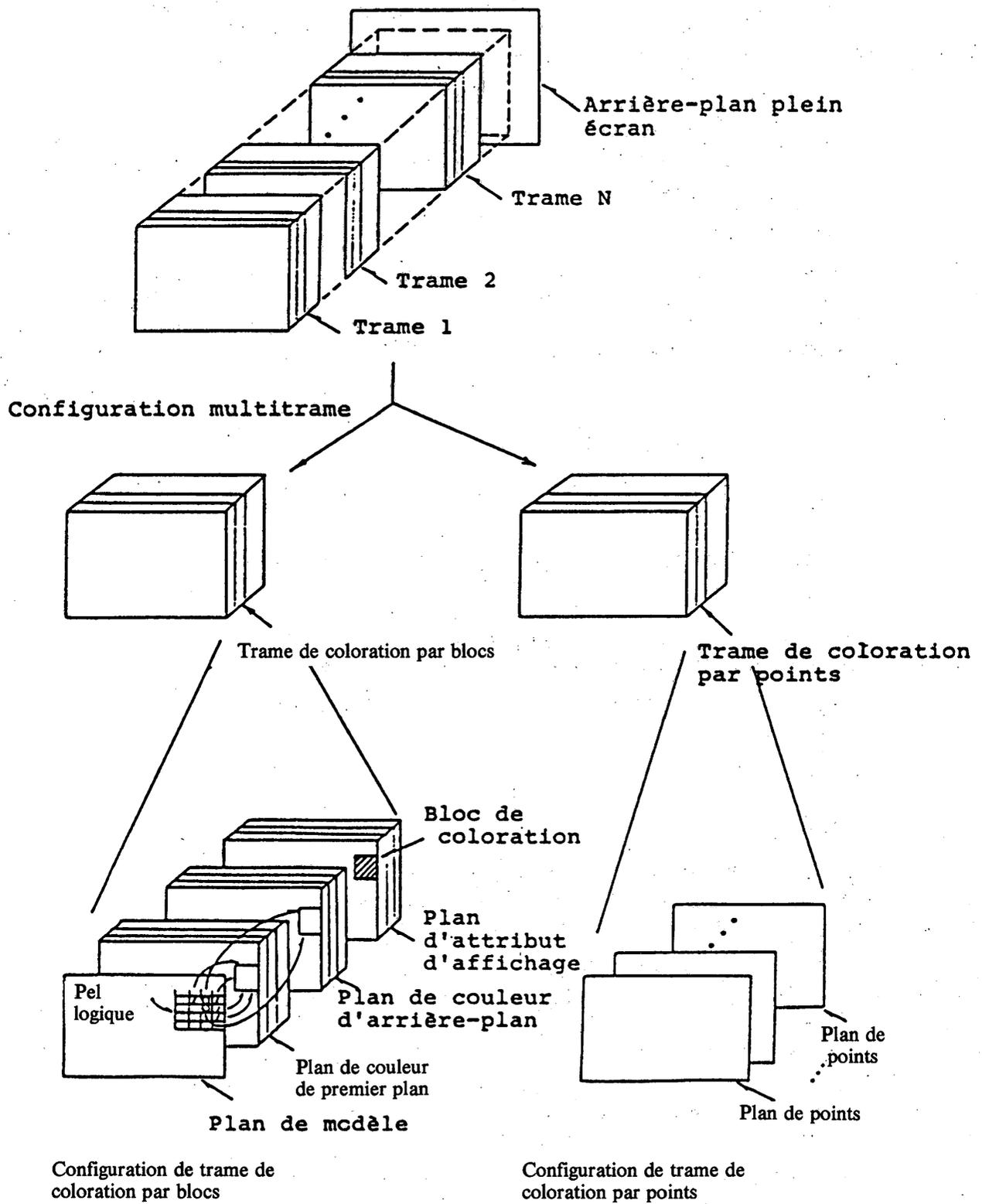


FIGURE 1.1

Concept de structure multitrame

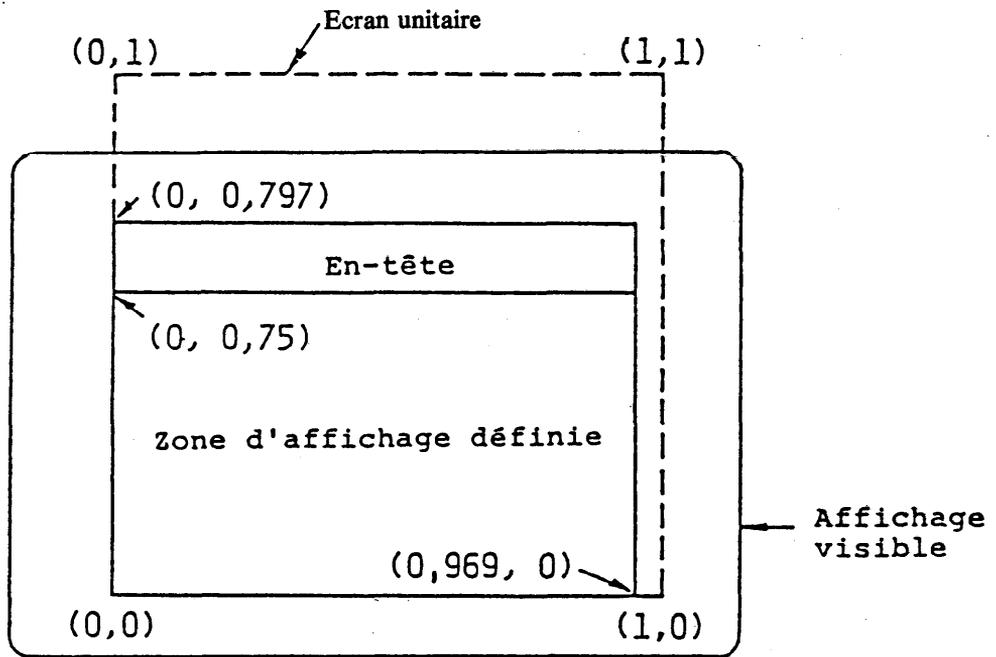


FIGURE 1.2

Concept d'écran unitaire

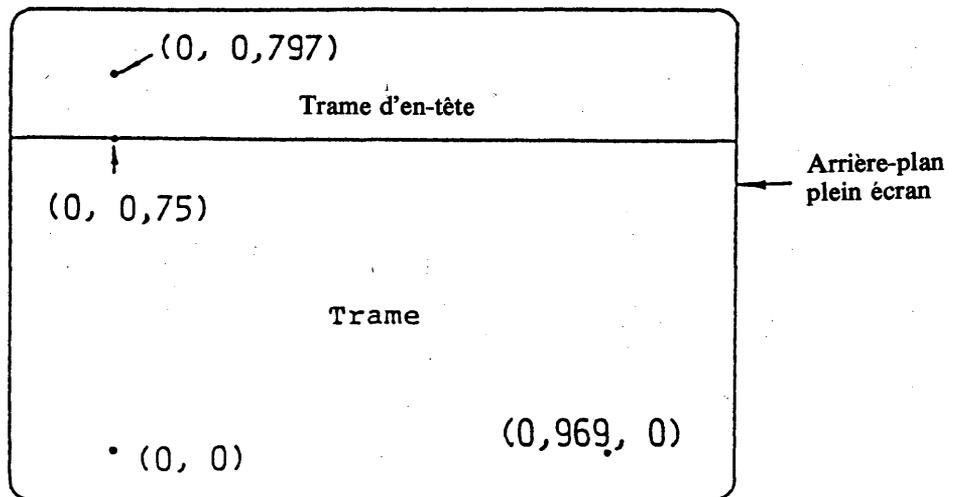


FIGURE 1.3

Structure d'arrière-plan plein écran

2.3 Structure de codage dans le mode code de caractère

La structure de codage dans le mode code de caractère se base sur les principes d'extension de code de la norme ISO 2022 pour les deux environnements 7 bits et 8 bits. La figure 2.2 représente la méthode d'extension de code de la syntaxe de données I. Les séquences d'échappement pour la désignation des ensembles-C et des ensembles-G sont indiquées dans le tableau 2.1.

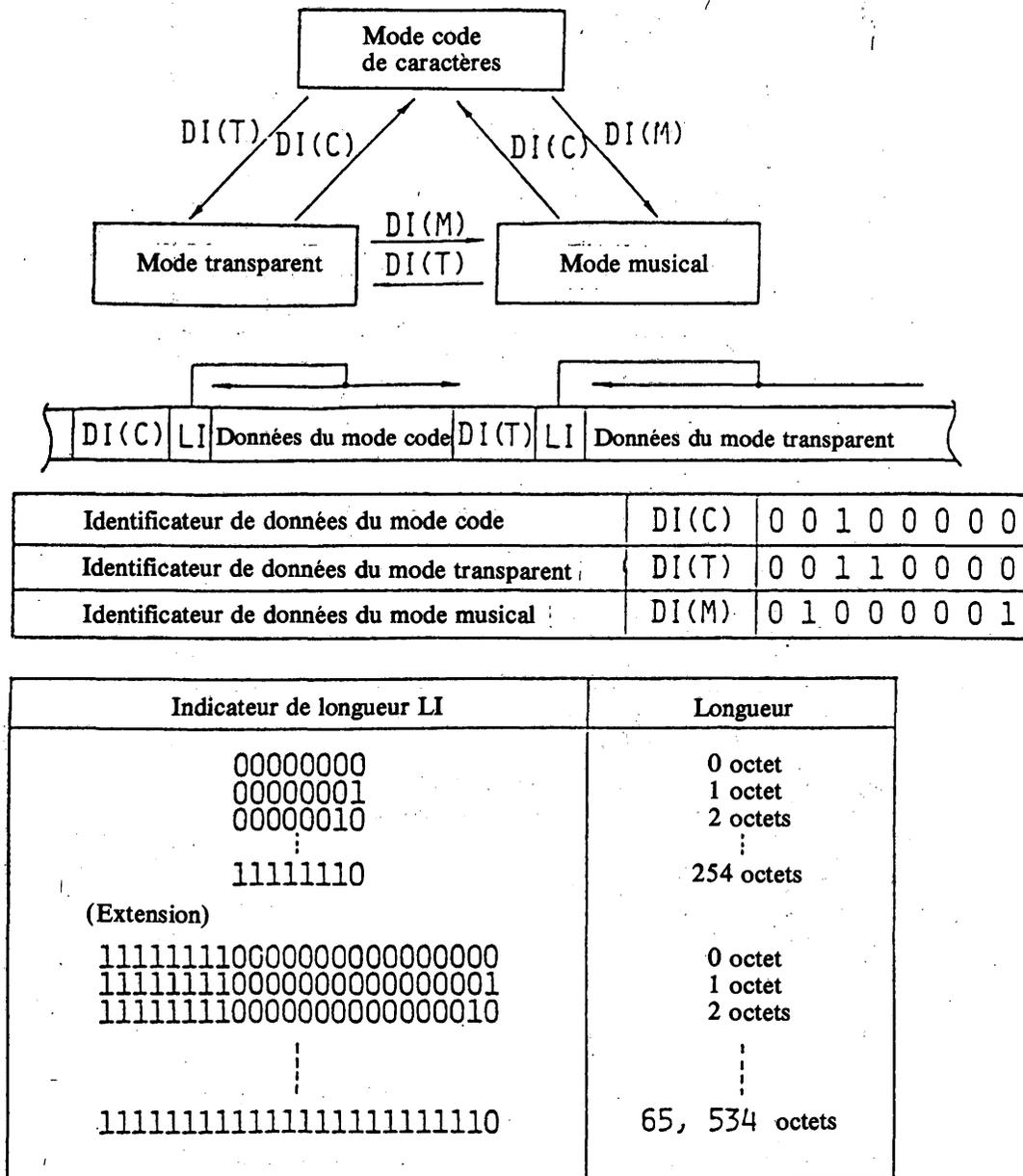


FIGURE 2.1

Concept de commutation de mode

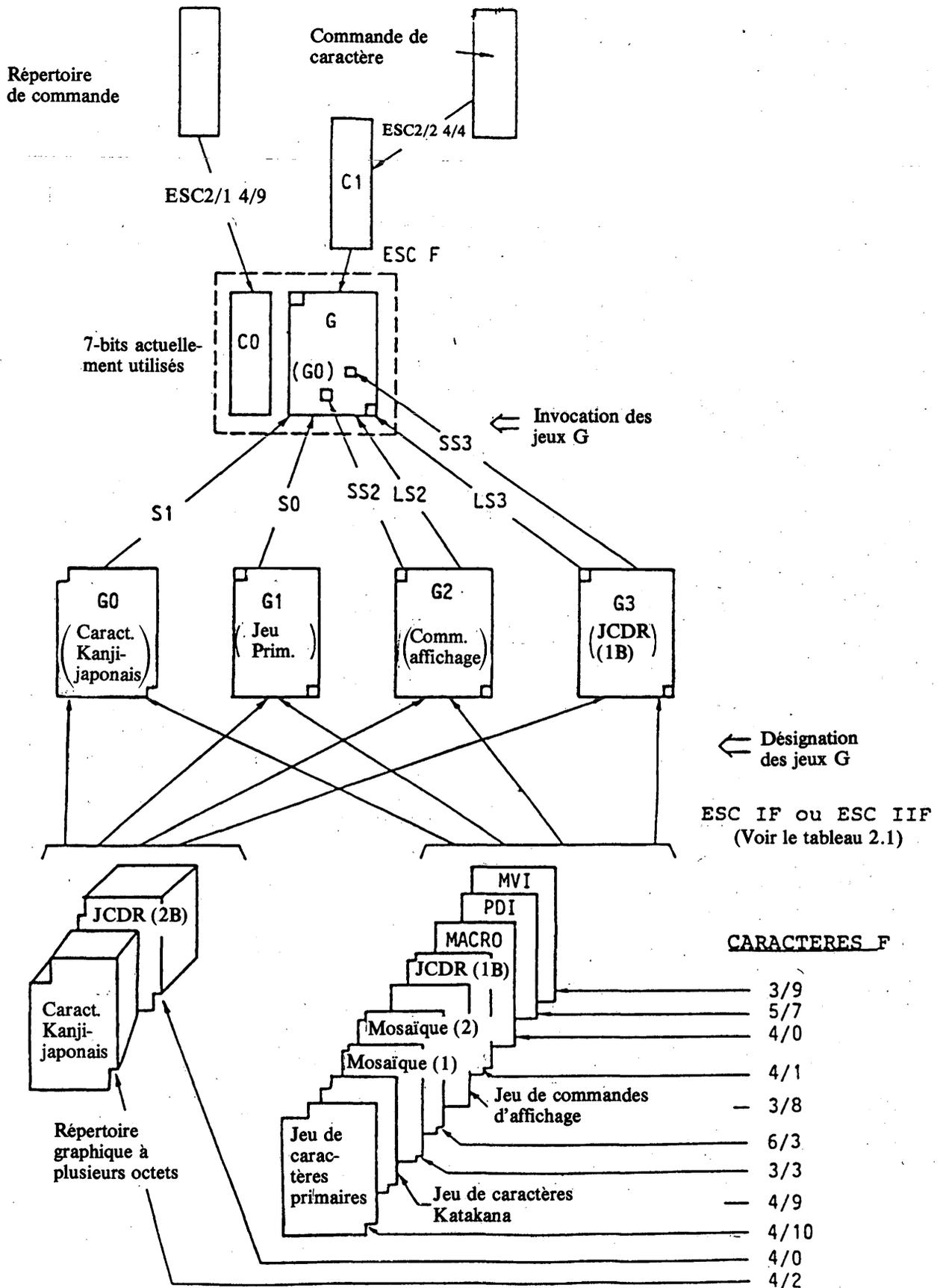


FIGURE 2.2 (1)

Extension de code dans le mode code de caractère (environnement 7 bits)

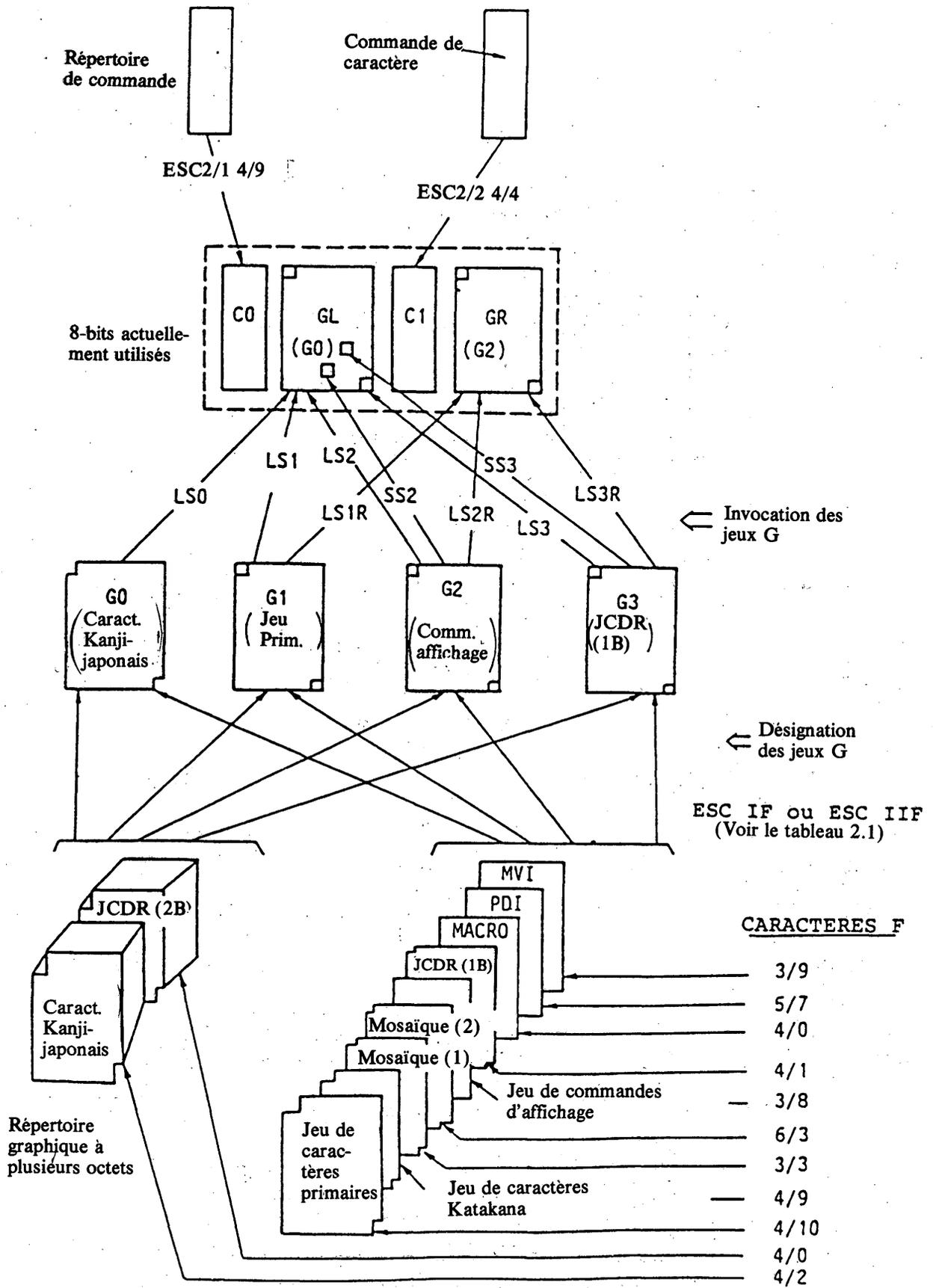


FIGURE 2.2 (2)

Extension de code dans le mode code de caractère (environnement 8 bits)

TABLEAU 2.1

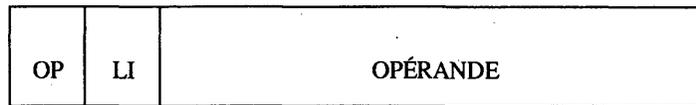
## Séquences d'échappement pour la désignation des jeux C et G

Séquence d'échappement	Jeu à désigner
Ensembles de commandes	
ESC 2/1 4/9	Jeu C0
ESC 2/2 4/4	Jeu C1

Jeu de caractères	G0	G1	G2	G3
Jeu de caractères primaires Jeu de caractères Katakana Jeu de mosaïques 1 Jeu de mosaïques 2	ESC 2/8 F	ESC 2/9 F	ESC 2/10 F	ESC 2/11 F
Jeu de commandes d'affichage Jeu PDI Jeu MVI	--	ESC 2/13 F	ESC 2/14 F	ESC 2/15 F
Jeu de caractères Kanji-japonais	ESC 2/4 F	ESC 2/4 2/9 F	ESC 2/4 2/10 F	ESC 2/4 2/11 F
Jeu de macros	--	ESC 2/13 2/0 F	ESC 2/14 2/0 F	ESC 2/15 2/0 F
Jeu JCDR 1	ESC 2/8 2/0 F	ESC 2/9 2/0 F	ESC 2/10 2/0 F	ESC 2/11 2/0 F
Jeu JCDR 2	ESC 2/4 2/8 2/0 F	ESC 2/4 2/9 2/0 F	ESC 2/4 2/10 2/0 F	ESC 2/4 2/11 2/0 F

Jeu de caractères	F
Jeu de caractères primaires	4/10
Jeu de caractères Katakana	4/9
Jeu de mosaïques 1	3/3
Jeu de mosaïques 2	6/3
Jeu de commandes d'affichage	3/8
Jeu PDI	5/7
Jeu MVI	3/9
Jeu de caractères Kanji-japonais	4/2
Jeu de macros	4/0
Jeu JCDR 1	4/1
Jeu JCDR 2	4/0

*Remarque* – L'ensemble de l'environnement de codage décrit dans la syntaxe de données I doit être désigné et appelé au moyen de la séquence d'échappement ESC 2/5 4/3, conformément à la norme 2022 de l'ISO (1982).



OP: Code d'opération  
LI: Indicateur de longueur

FIGURE 2.3

**Format d'unité de données photographiques**

2.4 *Structure de codage dans le mode transparent*

Dans le mode transparent, les données transparentes sont séparées en unités de données photographiques (PDU). La structure d'une PDU est indiquée sur la figure 2.3. Le premier octet (8 bits) d'une PDU est un code d'opération (opcode) qui définit le sens de la PDU. Le deuxième octet est un indicateur de longueur indiquant le nombre d'octets dans une PDU. Si l'expression décimale du deuxième octet est 255, les deux octets suivants (le troisième et le quatrième) indiquent la longueur réelle. Après l'indicateur de longueur, la séquence de données de bits nécessaire suit.

Le flux des données de niveau de présentation est illustré dans la figure 2.4.

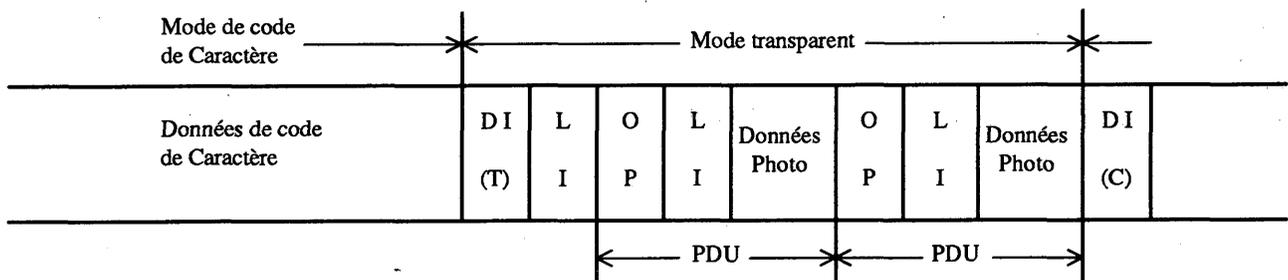


FIGURE 2.4

**Flux de données de niveau présentation (mode transparent)**

2.5 *Structure de codage dans le mode musical*

Les commandes et les paramètres sont codés en caractères dans le mode musical. Cependant, un jeu C et un jeu G différents de ceux du mode code de caractère sont réglés lors de l'invocation du mode.

Le flux de données de niveau présentation est illustré par la figure 2.5.

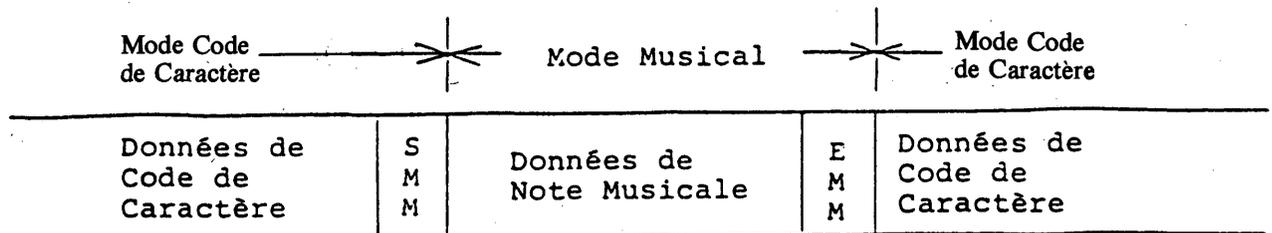


FIGURE 2.5

**Flux de données de niveau présentation (mode musical)**

### 3 Codage dans le mode code de caractère

#### 3.1 Ensemble de commandes C0

L'ensemble de commandes C0 est décrit dans le tableau 3.1. Les fonctions sont les suivantes:

TABLEAU 3.1  
Ensemble de commandes C0

				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) Le caractère NULL n'a pas d'effet sur le niveau de présentation. Il peut être utilisé pour le réglage de temps.
- BEL (0/7) Le caractère BELL est utilisé pour faire momentanément sonner une «cloche» pour attirer l'attention de l'utilisateur.
- CAN (1/8) CANCEL est utilisé pour terminer le traitement de tous les macros présentement en cours d'exécution. L'exécution est reprise au prochain caractère de niveau de présentation suivant l'appel macro terminé. L'effet de CAN est immédiat.
- APB (0/8) (ACTIVE POSITION BACKWARD) est utilisé pour positionner le curseur à une distance égale à l'espacement intercaractère parallèle au trajet des caractères dans la direction opposée au trajet des caractères. Si un tel mouvement implique le dépassement du bord de la zone d'affichage définie, alors le curseur est positionné au bord opposé de la zone d'affichage active et un APU est automatiquement exécuté.
- APF (0/9) (ACTIVE POSITION FORWARD) est utilisé pour positionner le curseur à une distance égale à l'espacement intercaractère parallèle au trajet des caractères dans la direction du trajet de caractère. Si un tel mouvement implique le dépassement du bord de la zone d'affichage définie, le curseur est positionné au bord opposé de la zone d'affichage définie et un APD est automatiquement exécuté.
- APD (0/10) (ACTIVE POSITION DOWN) est utilisé pour positionner le curseur à une distance égale à l'espacement interrangée perpendiculaire au trajet des caractères dans la direction perpendiculaire au trajet des caractères (- 90 deg.). Si un tel mouvement implique le dépassement du bord de la zone d'affichage définie, le curseur est positionné au bord opposé de la zone d'affichage définie.
- APU (0/11) (ACTIVE POSITION UP) est utilisé pour positionner le curseur à une distance égale à l'espacement interrangée perpendiculaire au trajet des caractères dans la direction perpendiculaire au trajet des caractères (90 deg.). Si un tel mouvement implique le dépassement du bord de la zone d'affichage définie, le curseur est positionné au bord opposé de la zone d'affichage définie.
- APR (0/13) (ACTIVE POSITION RETURN) est utilisé pour positionner le curseur à la position du premier caractère à l'intérieur de la zone d'affichage définie le long du trajet des caractères et un APD est automatiquement exécuté.
- APH (1/14) (ACTIVE POSITION HOME) est utilisé pour positionner le curseur à la position de caractère supérieur gauche dans la zone d'affichage définie.
- APS (1/12) (ACTIVE POSITION SET) est utilisé pour régler la position du curseur qui est spécifiée par le paramètre à deux octets suivant immédiatement un APS. Le premier octet représente l'adresse de la rangée et le deuxième octet représente l'adresse de la colonne. L'adresse est obtenue en prenant les valeurs binaires comprenant les bits b6 à b1, b6 étant le MSB. (Chaque octet correspond aux colonnes de 4 à 7.)
- Les opérations APS exactes sont les suivantes: premièrement, un APH est automatiquement exécuté. Ensuite, le curseur est positionné à N espacements interrangée dessous et à M espacements intercaractère vers la droite de la position initiale, N étant le nombre spécifié par le premier octet et M le nombre spécifié par le deuxième octet.
- Si l'un des deux octets se trouve en dehors des colonnes 4 à 7, le caractère APS et les deux octets suivants sont considérés comme des caractères NUL.
- NSR (1/15) NSR a deux fonctions. Premièrement, il réinitialise de manière non sélective tous les attributs d'affichage à leurs états de défaut (voir le § 6). Ensuite, il règle les positions de curseur au moyen du paramètre à deux octets suivant immédiatement le NSR. Le positionnement du curseur est identique à celui de APS, sauf que le APS positionne le curseur en fonction de l'espacement intercaractère et de l'espacement interrangée actuels, alors que NSR règle le curseur en fonction des valeurs de défaut. Les définitions MACRO et les définitions JCDR ne sont pas annulées par NSR.
- CS (0/12) CS réinitialise tous les attributs d'affichage à leurs états de défaut et active un APH automatique. Ensuite, toutes les définitions MACRO et les définitions DRCS sont annulées. CS efface tout l'écran.
- ESC (1/11) ESC introduit une séquence d'extension de code.
- LS0 (SI) (0/15) Glissement 0.
- LS1 (SO) (0/14) Glissement 1.

SS2 (1/9) Glissement unique 2.

SS3 (1/13) Glissement unique 3.

### 3.2 Ensemble de commandes C1

L'ensemble de commandes C1 est utilisé pour permettre la commande des attributs d'affichage de texte ainsi que pour déterminer les macros. Le tableau 3.2 décrit l'ensemble de commandes C1.

TABLEAU 3.2  
Ensemble de commandes C1

b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>			
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	MACRO*-P**
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

\* Ce code possède un paramètre

\*\* -P: Photographique

### 3.2.1 *Commandes de couleur*

Chaque caractère sur un écran possède une couleur de premier plan (couleur d'élément d'image où la valeur du plan de modèle est 1) et une couleur d'arrière-plan (couleur d'élément d'image où la valeur du plan de modèle est 0). Les deux couleurs (premier plan et arrière-plan) sont spécifiées au moyen d'une adresse d'entrée de table de consultation de couleur (LUT). Un nom de code est cependant dérivé de la couleur de défaut de la LUT.

Les couleurs de défaut comprennent six couleurs (rouge, vert, jaune, bleu, magenta, cyan) plus le noir et le blanc, et leurs versions d'intensité réduite (voir le tableau 3.5).

Une couleur de premier plan à pleine intensité est spécifiée par l'un des codes de commandes C1 suivants.

- BKF (Premier plan noir) Invoque le noir ou la 0<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- RDF (Premier plan rouge) Invoque le rouge ou la 1<sup>re</sup> couleur de la LUT comme couleur de premier plan.
- GRF (Premier plan vert) Invoque le vert ou la 2<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- YLF (Premier plan jaune) Invoque le jaune ou la 3<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- BLF (Premier plan bleu) Invoque le bleu ou la 4<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- MGF (Premier plan magenta) Invoque le magenta ou la 5<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- CNF (Premier plan cyan) Invoque le cyan ou la 6<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- WHF (Premier plan blanc) Invoque le blanc ou la 7<sup>e</sup> couleur de la LUT comme couleur de premier plan.

Une couleur d'arrière-plan à pleine intensité, une couleur de premier plan d'intensité réduite ou une couleur d'arrière-plan d'intensité réduite sont spécifiées par un paramètre d'un octet suivant le code de commande COL C1.

- COL 4/8 – TRF (Premier plan transparent) Invoque le transparent ou la 8<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/9 – RDFR (Premier plan rouge réduit) Invoque un rouge d'intensité réduite ou la 9<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/10 – GRFR (Premier plan vert réduit) Invoque un vert d'intensité réduite ou la 10<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/11 – YLFR (Premier plan jaune réduit) Invoque un jaune d'intensité réduite ou la 11<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/12 – BLFR (Premier plan bleu réduit) Invoque un bleu d'intensité réduite ou la 12<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/13 – MGFR (Premier plan magenta réduit) Invoque un magenta d'intensité réduite ou la 13<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/14 – CNFR (Premier plan cyan réduit) Invoque un cyan d'intensité réduite ou la 14<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 4/15 – WHFR (Premier plan blanc réduit) Invoque un blanc d'intensité réduite ou la 15<sup>e</sup> couleur de la LUT comme couleur de premier plan.
- COL 5/0 – BKB (Arrière-plan noir) Invoque le noir ou la 0<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/1 – RDB (Arrière-plan rouge) Invoque le rouge ou la 1<sup>re</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/2 – GRB (Arrière-plan vert) Invoque le vert ou la 2<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/3 – YLB (Arrière-plan jaune) Invoque le jaune ou la 3<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/4 – BLB (Arrière-plan bleu) Invoque le bleu ou la 4<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.

- COL 5/5 – MGB (Arrière-plan magenta) Invoque le magenta ou la 5<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/6 – CNB (Arrière-plan cyan) Invoque le cyan ou la 6<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/7 – WHB (Arrière-plan blanc) Invoque le blanc ou la 7<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/8 – TRB (Arrière-plan transparent) Invoque le transparent ou la 8<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/9 – RDBR (Arrière-plan rouge réduit) Invoque un rouge d'intensité réduite ou la 9<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/10 – GRBR (Arrière-plan vert réduit) Invoque un vert d'intensité réduite ou la 10<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/11 – YLBR (Arrière-plan jaune réduit) Invoque un jaune d'intensité réduite ou la 11<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/12 – BLBR (Arrière-plan bleu réduit) Invoque un bleu d'intensité réduite ou la 12<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/13 – MGBR (Arrière-plan magenta réduit) Invoque un magenta d'intensité réduite ou la 13<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/14 – CNBR (Arrière-plan cyan réduit) Invoque un cyan d'intensité réduite ou la 14<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.
- COL 5/15 – WHBR (Arrière-plan blanc réduit) Invoque un blanc d'intensité réduite ou la 15<sup>e</sup> couleur de la LUT comme couleur d'arrière-plan.

Une couleur de premier plan et une couleur d'arrière-plan spécifiées restent inchangées jusqu'à ce qu'une nouvelle couleur soit spécifiée. L'exécution de NSR ou CS réinitialise également la spécification de couleur.

La couleur de premier plan de défaut est le blanc et la couleur d'arrière-plan de défaut est le transparent.

### 3.2.2 *Commandes de taille de caractère*

Les codes suivants définissent une taille de caractère à utiliser. La taille de caractère, une fois définie, reste inchangée jusqu'à ce qu'elle soit altérée par l'ensemble de commandes C1 suivant, une instruction de commande d'affichage (TEXTE P<sup>2)</sup>), NSR ou CS.

- NSZ (Taille normale) La largeur et la hauteur des caractères sont les mêmes que celles du bloc de caractères spécifié par la commande TEXTE P qui est décrite plus loin.
- SSZ (Petite taille) La largeur et la hauteur des caractères sont réduites à la moitié de celles de NSZ.
- MSZ (Taille moyenne) La largeur des caractères est réduite à la moitié de celle de NSZ. La hauteur des caractères est la même que celle de NSZ.
- SZX (Commande de taille) SZX est le code de commandes C1 qui spécifie une taille de caractère DBH, DBW et DBS par un paramètre d'un octet suivant le code SZX.
- SZX 4/1 – DBH (Hauteur double): La hauteur des caractères est le double de celle de NSZ. La largeur des caractères est la même que celle de NSZ.
- SZX 4/4 – DBW (Largeur double): La largeur des caractères est le double de celle de NSZ. La hauteur des caractères est la même que celle de NSZ.
- SZX 4/5 – DBS (Taille double): La largeur et la hauteur des caractères sont le double de celles de NSZ.

Les dimensions de défaut de la taille de caractère sont «normal».

---

<sup>2)</sup> P signifie «photo».

### 3.2.3 *Commandes de curseur*

CON (Curseur en marche) Le curseur est affiché.

COF (Curseur à l'arrêt) Le curseur n'est pas affiché.  
L'état de défaut est curseur à l'arrêt.

### 3.2.4 *Commandes de clignotement*

Le clignotement est un procédé dans lequel une couleur de premier plan est alternativement remplacée par une couleur d'arrière-plan. L'attribut de clignotement peut être placé sur la base de taille de caractère ou sur la base de bloc de coloration. L'état de défaut est fixe.

Le code FLC (commande de clignotement), qui est l'un des codes de commande C1, spécifie les attributs de clignotement au moyen d'un paramètre d'un octet suivant le code FLC.

FLC 4/0 – Clignotement normal: Applique les attributs de clignotement normaux (50%).

FLC 4/7 – Clignotement inversé: Applique les attributs de clignotement inversés.

FLC 4/4 – Clignotement triphasique 1: Applique la 1<sup>re</sup> phase de clignotement triphasique.

FLC 4/2 – Clignotement triphasique 2: Applique la 2<sup>e</sup> phase de clignotement triphasique.

FLC 4/1 – Clignotement triphasique 3: Applique la 3<sup>e</sup> phase de clignotement triphasique.

FLC 4/3 – Clignotement inversé triphasique 1: Applique la 1<sup>re</sup> phase inversée de clignotement triphasique.

FLC 4/5 – Clignotement inversé triphasique 2: Applique la 2<sup>e</sup> phase inversée de clignotement triphasique.

FLC 4/6 – Clignotement inversé triphasique 3: Applique la 3<sup>e</sup> phase inversée de clignotement triphasique.

FLC 4/15 – Fixe: Annule l'application de tous les attributs de clignotement.

La séquence FLC 4/0, A, B, C, D, FLC 4/15 entraîne un clignotement normal des caractères «ABCD».

### 3.2.5 *Commandes de répétition*

Le code RPC (commande de répétition) entraîne l'affichage du caractère de jeu G suivant transmis, si le caractère suivant est un caractère sans espacement, le caractère sans espacement et le caractère suivant seront affichés tous les deux un nombre de fois spécifié par l'octet suivant. L'octet doit être des colonnes 4 à 7. Le comptage de répétitions est donné par le nombre binaire comprenant les bits b6 à b1, b6 étant le MSB. Si l'octet suivant le code RPC ne provient pas des colonnes 4 à 7, le code RPC n'est pas exécuté. RPC 4/0 a un sens spécial causant la répétition jusqu'à la fin de la ligne.

La séquence A, RPC 4/3, B, C entraîne l'affichage de «A BBB C».

### 3.2.6 *Commandes de traçage de ligne*

Les codes suivants commandent l'application de l'attribut de traçage de ligne.

STL (Début de traçage de ligne) Applique l'attribut de traçage de ligne: Les caractères sont affichés avec soulignement alors que les modèles de mosaïque sont affichés au moyen d'une police séparée.

SPL (Arrêt de traçage de ligne) Réinitialise l'attribut de traçage de ligne.

### 3.2.7 *Commandes de dissimulation*

Les codes CDC (commande d'affichage dissimulé) commandent l'attribut d'affichage dissimulé. Lorsque cet attribut est appliqué, les caractères et JCDR deviennent invisibles. La position de caractère prend une avance ordinaire. Les caractères dissimulés sont rendus visibles par une action spécifique de l'utilisateur. Le code CDC, qui est l'un des codes de commande C1, spécifie les attributs d'affichage dissimulé par un paramètre d'un octet suivant le code CDC.

CDC 4/0 – CDY (Affichage dissimulé) Applique l'attribut d'affichage dissimulé aux caractères suivants.

CDC 4/15 – SCD (Arrêt d'affichage dissimulé) Réinitialise l'attribut d'affichage dissimulé.

La séquence A, B, C, CDC 4/0, D, E, F, ..., CDC 4/15 entraîne l'affichage sur un terminal de «ABC» tout d'abord et «DEF ...» est rendu visible par l'action de l'utilisateur.

### 3.2.8 Commandes de protection

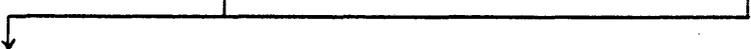
Des blocs de caractères protégés ne peuvent être ni altérés, ni manipulés, ni effacés par action de l'utilisateur. Tout l'écran est protégé par défaut.

UNP (Non protégé) Les caractères suivants ne seront pas protégés.

PRT (Protégé) Les caractères suivants ne seront pas protégés.

Exemple:

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



Ces caractères ne sont pas protégés alors que les autres caractères le sont.

### 3.2.9 Instruction macro photo

#### 3.2.9.1 Généralités

Le service complémentaire d'instruction macro photo permet d'enregistrer en mémoire dans le terminal une chaîne de caractères du niveau présentation, y compris des instructions de mode transparent et de les exécuter ultérieurement par un simple appel macro photo. Un nom de macro photo est l'un des 96 caractères (de 2/0 à 7/15), désignés PM0 à PM95.

#### 3.2.9.2 DEF MACRO P

La commande DEF MACRO P est spécifiée par le paramètre 4/0 suivant le code MACRO P qui est l'un des codes de commandes C1. L'instruction DEF MACRO P est utilisée pour définir une instruction macro photo. Le caractère suivant le code est le nom de l'instruction macro photo. Tous les codes de caractère et les séquences de bits de mode transparent sont enregistrés en mémoire dans le terminal sous le nom précis de la macro photo. La définition de l'instruction de macro photo se termine lors de la réception des instructions DEF MACRO P, DEFP MACRO P, DEFT MACRO P et FIN DE MACRO P.

Les caractères et instructions suivant le caractère de nom macro photo ne sont pas exécutés au moment de la réception. Ni le code de terminaison ni son code ESC précédent (dans un environnement 7 bits) ne sont enregistrés en mémoire comme partie de l'instruction macro photo.

La définition d'une instruction photo remplace toute instruction macro photo existant préalablement sous le même nom. S'il n'y a pas de caractères ou d'instructions valides entre le nom macro photo et le code de terminaison (y compris son code ESC précédent), la commande macro photo est éliminée. Toutes les macros photo sont annulées simultanément par l'instruction RÉINITIALISER P ou CS.

#### 3.2.9.3 DEFP MACRO P

L'instruction DEFP MACRO P est spécifiée par le paramètre 4/1 suivant le code de commande MACRO P C1.

Cette instruction est identique à l'instruction DEF MACRO P sauf qu'elle enregistre en mémoire et exécute simultanément les caractères entrants qui forment la MACRO.

#### 3.2.9.4 DEFT MACRO P

L'instruction DEFT MACRO P est spécifiée par le paramètre 4/2 suivant le code de commande C1 MACRO P.

Cette instruction est utilisée pour définir des macros d'émission. Lorsqu'elles sont appelées, ces macros ne sont pas exécutées, mais retransmises à l'ordinateur central.

### 3.2.9.5 FIN DE MACRO P

L'instruction FIN DE MACRO P est spécifiée par le paramètre 4/15 suivant le code de commande C1.

Cette instruction termine les opérations DEF MACRO P, DEFP MACRO P ou DEFT MACRO P en cours.

## 3.3 Ensemble d'instructions de commandes d'affichage

### 3.3.1 Généralités

L'ensemble d'instructions de commande d'affichage permet de commander les attributs d'affichage, le format des données de niveau présentation et la définition de la structure d'affichage. Les instructions de commande d'affichage ont un effet sur les informations sous forme de texte, les informations photographiques et les informations géométriques.

L'ensemble d'instructions de commande d'affichage est l'un des jeux G. Une instruction de commande d'affichage se compose d'un code d'opération d'un seul octet et, si nécessaire, d'un ou plusieurs opérandes. Chaque opérande consiste en un ou plusieurs octets de données numériques ou de combinaison de bits:

- i) format fixe,
- ii) format à valeur unique,
- iii) format à valeur multiple.

Les opérandes à format fixe consistent en un ou plusieurs octets de données numériques ou de combinaison de bits dont la longueur et l'interprétation dépendent du code d'opération avec lequel ils sont utilisés.

Les opérandes de format à valeur unique consistent en un, deux, trois ou quatre octets de données numériques, de la manière déterminée par l'instruction DOMAINE P décrite plus loin. Ils sont interprétés comme entiers non signés composés de la séquence de bits concaténés pris consécutivement (du bit de poids fort ou b<sub>6</sub> au bit de poids faible ou b<sub>1</sub>) à partir des octets de données numériques de la manière indiquée sur la figure 3.1 (1).

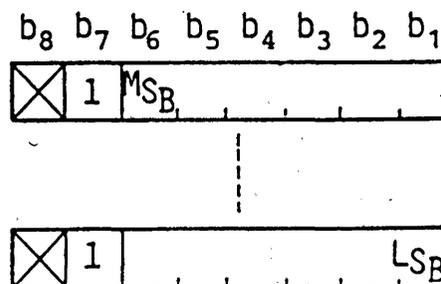


FIGURE 3.1 (1)

Format d'opérande à valeur unique

Les opérandes à valeur multiple consistent en un à huit octets de données numériques, de la manière déterminée par l'instruction DOMAINE P. Ces opérandes sont utilisés pour spécifier les informations de coordonnées, lorsqu'elles servent à spécifier des dimensions ou les informations de couleur, lorsqu'elles sont utilisées avec l'instruction INITIALISATION DE LUT décrite plus loin.

Tous ces opérandes de coordonnées sont interprétés comme des nombres compléments de deux signés, le MSB représentant 0,5.

Lorsqu'un opérande à valeur multiple est utilisé avec la commande INITIALISATION DE LUT, il spécifie une valeur de couleur non signée dans le système de couleur RVB (rouge-vert-bleu). La représentation de valeur de couleur dans le cadre de l'opérande à valeur multiple est indiquée dans la figure 3.1 (2), où la valeur de couleur est donnée par un décimal binaire non signé. La valeur de couleur «0» indique l'intensité la plus faible, alors que la valeur «1» indique l'intensité la plus forte.

Le tableau 3.3 indique les types d'opérandes utilisés par chacun des codes d'opération.

L'ensemble complet des instructions de commande d'affichage fait l'objet du tableau 3.4.

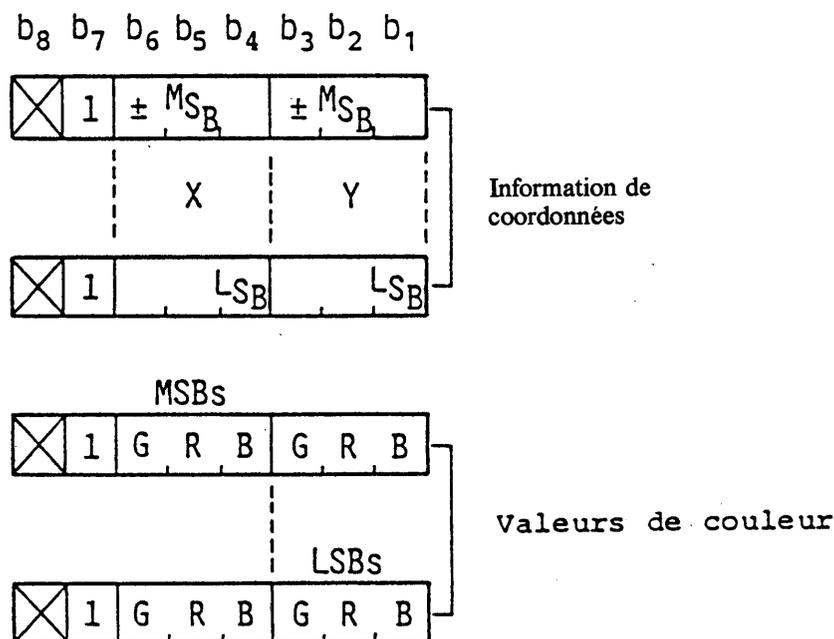


FIGURE 3.1 (2)

Format d'opérande à valeur multiple

TABLEAU 3.3

Types d'opérande

Code d'opération	Type d'opérande
RÉINITIALISATION P	Fixe 1 octet
DOMAINE P	Fixe 2 octets
PEL LOGIQUE	Valeur multiple
MODE D'AFFICHAGE	Fixe 1 octet
TEXTE P	Fixe 2 octets/Valeur multiple
ATTENTE P	Fixe 1 octet
TRAME	Valeur multiple
TRAME D'EN-TÊTE	Valeur multiple
INITIALISATION DE LUT	Valeur unique/Valeur multiple
CLIGNOTEMENT P	Valeur unique/Fixe 3 octets
ZONE	Valeur multiple
INITIALISATION DE TRAME	Valeur unique
ASSIGNATION DE TRAME	Valeur unique

TABLEAU 3.4

Ensemble d'instructions de commande d'affichage

					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	Réinitialisation-P						
0	0	1	0	2	Domaine-P						
0	0	1	1	3	Pei logique						
0	1	0	0	4	Mode d'affichage	Texte-P					
0	1	0	1	5	Zone						
0	1	1	0	6	Initialisation de trame						
0	1	1	1	7	Assignment de trame						
1	0	0	0	8		Trame					
1	0	0	1	9		Trame d'en-tête					
1	0	1	0	10		Initialisation de LUT					
1	0	1	1	11							
1	1	0	0	12	Attente-P	Clignotement-P					
1	1	0	1	13							
1	1	1	0	14							
1	1	1	1	15							

Données numérique



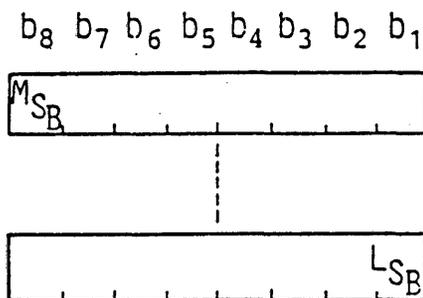


FIGURE 3.3 (1)

**Opérande d'instruction de mode transparent à valeur unique**

ii) *Longueur d'opérande à valeur multiple*

b5	b4	b3	Longueur d'opérande à valeur multiple (octets)
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

\*\* Valeur de défaut pour les instructions de mode transparent.

\*\*\* Valeur de défaut pour les instructions de mode code de caractère.

[Le format d'opérande de commande de mode transparent à valeur multiple est indiqué sur la figure 3.3 (2).]

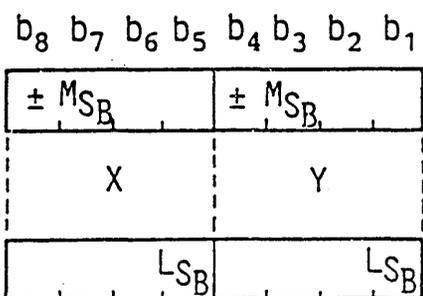


FIGURE 3.3 (2)

**Opérande d'instruction de mode transparent à valeur multiple**

### iii) *Dimensions*

Le bit 6 de chaque octet d'opérande détermine les dimensions de la spécification de coordonnées. «0» indique le mode à deux dimensions et «1» indique le mode à trois dimensions, cependant, la définition du mode à trois dimensions est réservée pour une normalisation future. En conséquence, le bit 6 doit toujours être «0» jusqu'à la normalisation finale.

### iv) *Longueur d'opérande*

Si la longueur d'un opérande suivant un code d'opération d'instruction de mode code de caractère est inférieure à la longueur préalablement spécifiée par l'instruction DOMAINE P (ou la longueur implicite dans le cas du format fixe), alors le reste de l'opérande est rempli de zéros, sauf indication contraire dans la définition de l'instruction. Si la longueur d'un opérande suivant un code d'opération d'instruction de mode code de caractère est supérieure à la longueur préalablement spécifiée par l'instruction DOMAINE P (ou la longueur implicite), elle est considérée comme indication pour répéter l'exécution de l'instruction, les données numériques suivantes étant prises comme nouvel opérande, sauf indication contraire.

#### 3.3.2.2 *PEL LOGIQUE*

L'instruction PEL LOGIQUE définit la taille du pel logique et du bloc de coloration. L'instruction PEL LOGIQUE prend deux opérandes à valeur multiple comme indiqué sur la figure 3.4. Le premier opérande spécifie la taille du pel logique. La largeur (dx) et la hauteur (dy) d'un pel logique sont interprétées comme valeurs de coordonnées relatives.

Le deuxième opérande à valeur multiple spécifie la taille du bloc de coloration. La taille du bloc de coloration est la même pour le plan de couleur de premier plan, le plan de couleur d'arrière-plan et le plan d'attribut d'affichage, et est normalement supérieure à celle du pel logique.

#### 3.3.2.3 *Commandes du tableau de consultation des couleurs (LUT)*

L'instruction INITIALISATION DE LUT et l'instruction CLIGNOTEMENT P permettent de commander le tableau de consultation des couleurs (LUT). Le LUT de trame de coloration par blocs est généralement utilisé pour les deux couleurs de premier plan et d'arrière-plan.

##### 3.3.2.3.1 *INITIALISATION DE LUT*

L'instruction INITIALISATION DE LUT est utilisée pour charger les valeurs de couleur RVB dans le LUT. Elle prend un opérande à valeur unique suivi d'un opérande à valeur multiple, comme indiqué sur la figure 3.5. L'opérande à valeur unique spécifie l'entrée de LUT qui est justifiée à gauche dans l'opérande. L'opérande à valeur multiple spécifie les valeurs de couleur pour l'entrée assignée. Si des données numériques supplémentaires suivent cet opérande, l'instruction INITIALISATION DE LUT est implicitement répétée, l'adresse d'entrée de LUT étant automatiquement incrémentée. Le processus d'incrémation débute du côté MSB.

S'il n'y a pas d'opérande à valeur multiple, chaque valeur de R (rouge), V (vert) et B (bleu) est considérée comme «0», c'est-à-dire transparent.

Le contenu de défaut du LUT est indiqué dans le tableau 3.5.

##### 3.3.2.3.2 *CLIGNOTEMENT P*

L'instruction CLIGNOTEMENT P déclenche un processus de clignotement à l'entrée du LUT. Le processus de clignotement recouvre périodiquement le contenu de l'entrée spécifiée (la couleur «clignotement en provenance de») et substitue le contenu d'une autre entrée dans le LUT (la couleur «clignotement à destination de»).

L'instruction CLIGNOTEMENT P est accompagnée de deux opérandes à valeur unique suivis par trois opérandes à format fixe. Le premier opérande à valeur unique est la spécification de couleur «clignotement en provenance de», spécifiée comme une adresse d'entrée du LUT. Le deuxième opérande à valeur unique spécifie la couleur «clignotement à destination de».

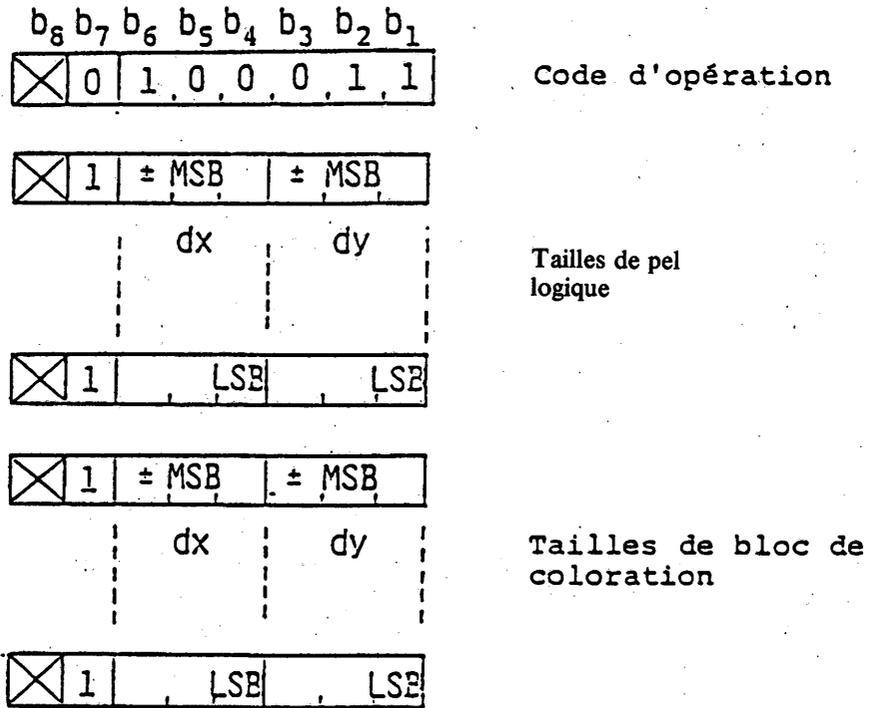


FIGURE 3.4  
PEL LOGIQUE

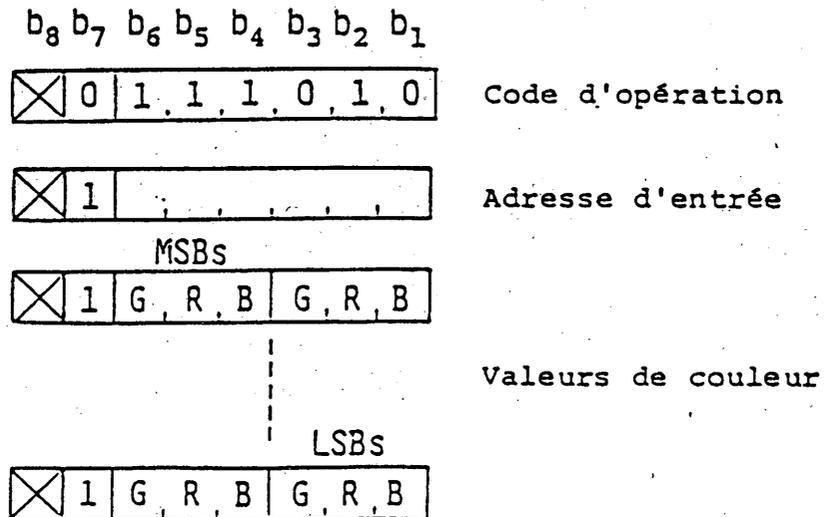


FIGURE 3.5  
INITIALISATION DE LUT

TABLEAU 3.5

Tableau de consultation des couleurs

	Adresse d'entrée (IR, B,V,R)	Valeurs de couleur			Couleur nominale
		B	V	R	
0	0 0 0 0	0001	0001	0001	Noir
1	0 0 0 1	0000	0000	1111	Rouge
2	0 0 1 0	0000	1111	0000	Vert
3	0 0 1 1	0000	1111	1111	Jaune
4	0 1 0 0	1111	0000	0000	Bleu
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	Blanc
8	1 0 0 0	0000	0000	0000	Transparent
9	1 0 0 1	0000	0000	0111	IR Rouge
10	1 0 1 0	0000	0111	0000	IR Vert
11	1 0 1 1	0000	0111	0111	IR Jaune
12	1 1 0 0	0111	0000	0000	IR Bleu
13	1 1 0 1	0111	0000	0111	IR Magenta
14	1 1 1 0	0111	0111	0000	IR Cyan
15	1 1 1 1	0111	0111	0111	Gris

IR: Intensité réduite

Le premier opérande à format fixe suivant le deuxième opérande à valeur unique spécifie l'intervalle «MARCHE» qui est la période pendant laquelle la couleur «clignotement à destination de» est activée. Le deuxième opérande à format fixe spécifie l'intervalle «ARRÊT» qui est la période pendant laquelle la couleur «clignotement à destination de» est activée. Le troisième opérande à format fixe spécifie le retard de phase. Le retard de phase se rapporte au début de l'intervalle «MARCHE» du processus actif le plus récemment défini. Ces valeurs sont données en unités de  $1/10^6$  de seconde.

La figure 3.6 indique le format de l'instruction CLIGNOTEMENT P

#### 3.3.2.4 Commandes de couleur de trame

L'instruction TRAME et l'instruction TRAME D'EN-TÊTE spécifient respectivement la couleur de la trame et la couleur de la trame d'en-tête. Les deux instructions ont le même format (voir les figures 3.7 et 3.8). Les valeurs de couleur sont spécifiées par un opérande à valeur multiple. Une fois réglée, la couleur de la trame ou la couleur de la trame d'en-tête reste inchangée jusqu'à ce qu'une autre instruction TRAME, une autre instruction TRAME D'EN-TÊTE ou un CS soit exécutée. Les valeurs de couleur de défaut des couleurs de trame et d'en-tête sont  $R = V = 0$  et  $B = 0,0111$  - - - (=  $1/2$ ; couleur bleu réduit).

#### 3.3.2.5 RÉINITIALISATION P

L'instruction RÉINITIALISATION P est utilisée pour réinitialiser sélectivement les paramètres de la manière décrite ci-dessous. Elle nécessite un opérande à format fixe d'un octet unique (voir la figure 3.9).

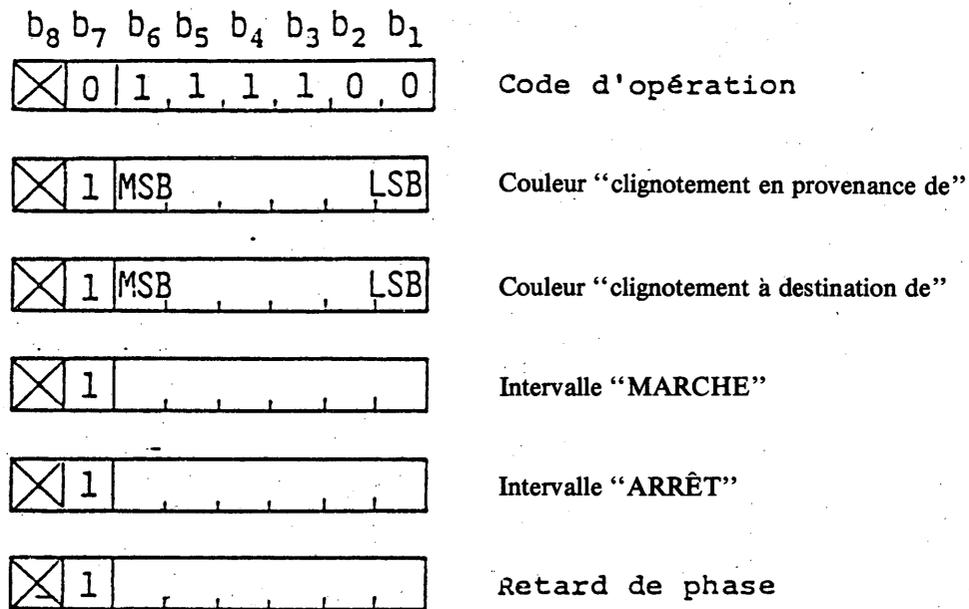


FIGURE 3.6  
CLIGNOTEMENT P

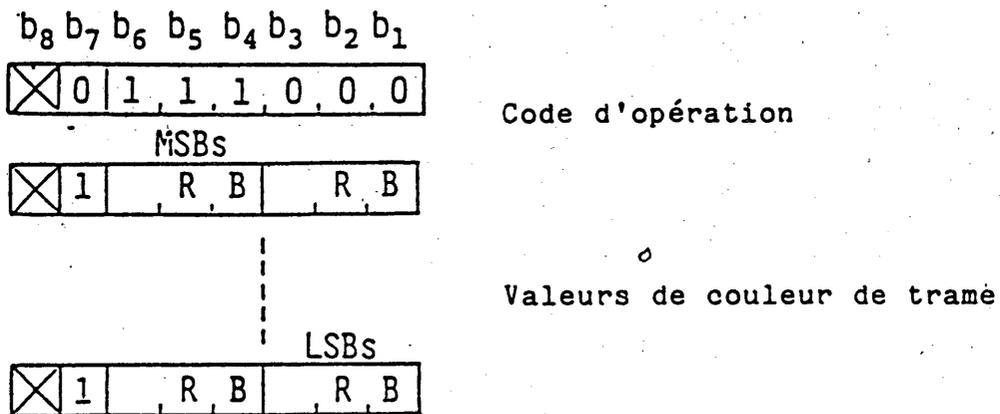


FIGURE 3.7  
TRAME

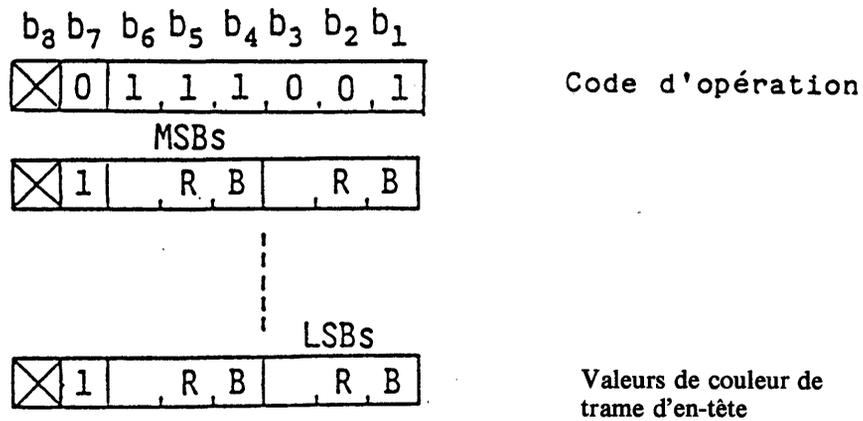


FIGURE 3.8  
TRAME D'EN-TÊTE

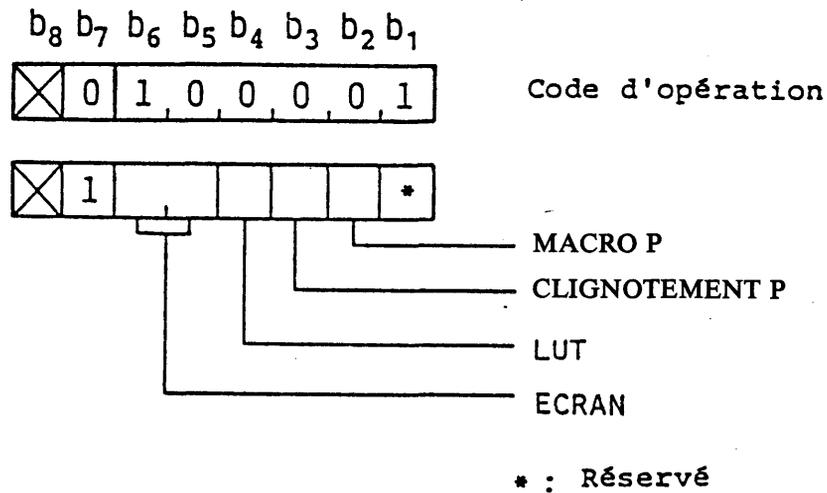


FIGURE 3.9  
RÉINITIALISATION P

- |    |    |  |
|----|----|--|
| b6 | b5 | Ecran  |
| 0  | 0  | Pas d'opération  |
| 0  | 1  | Tout le plan de modèle est réinitialisé à 0. La couleur du premier plan entier est réglée à la 7 <sup>e</sup> couleur du LUT (couleur de défaut; blanc), alors que la couleur de l'arrière-plan entier est réglée à la 8 <sup>e</sup> couleur du LUT (couleur de défaut; transparent). |

Lorsque l'écran est effacé (autre que  $b_6 = b_5 = 0$ ), tous les attributs de clignotement, de dissimulation et de protection sont annulés.

- b4 LUT
  - 0 Pas d'opération
  - 1 Le LUT est réglé aux couleurs de défaut
- b3 CLIGNOTEMENT P
  - 0 Pas d'opération
  - 1 Tous les processus de clignotement sont terminés
- b2 MACRO P
  - 0 Pas d'opération
  - 1 Toutes les macros photo sont annulées

### 3.3.2.6 MODE D'AFFICHAGE

L'instruction MODE D'AFFICHAGE spécifie les attributs d'affichage. Le code d'opération du MODE D'AFFICHAGE est accompagné d'un opérande à format fixe d'un octet unique dont les bits (de  $b_4$  à  $b_6$ ) commandent les différents attributs (voir la figure 3.10).

- b6 Défilement
  - 0 Défilement (arrêt)
  - 1 Défilement (marche)

L'état de défaut est défilement arrêté.

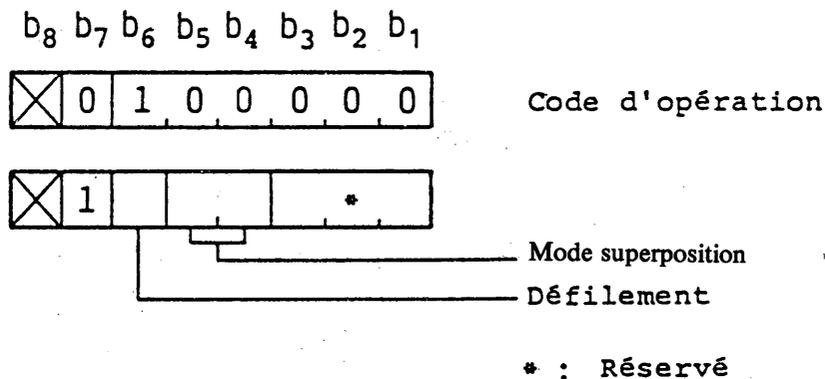


FIGURE 3.10  
MODE D'AFFICHAGE

Les opérations de mode de défilement marche et de mode de défilement arrêt sont les suivantes:

- i) Mode de défilement arrêt
  - Dans le cas d'une instruction photographique  
Les données sont écrites au point spécifié par l'instruction.
  - Dans le cas d'un texte  
Lorsque la position du caractère suivant dépasse la limite inférieure de l'écran, le caractère suivant sera affiché au coin supérieur gauche de l'écran.

ii) Mode de défilement marche

– Dans le cas d'une instruction photographique

Lorsque le mode de défilement marche est activé par l'instruction MODE D'AFFICHAGE, un registre pour la mise en mémoire d'une valeur de coordonnée Y (registre de coordonnée Y) est tout d'abord pré-réglé à 0,75. Lors de la réception de nouvelles données, tout le contenu de l'écran défile vers le haut sur une distance donnée comme la différence entre le contenu du registre de coordonnée Y et la valeur de coordonnée Y indiquée par l'instruction, et en même temps position Y = 0. Si la différence devient 0,75, elle est considérée comme 0. Après cela, la valeur de coordonnée Y indiquée par l'instruction est réglée dans le registre de coordonnée Y. Lorsqu'une instruction photographique indique d'écrire les données bloc par bloc plutôt que ligne par ligne et que la position du nouveau bloc de champ dépasse la limite inférieure de l'écran, tout le contenu de l'écran défile vers le haut de manière que le bas du nouveau bloc de champ corresponde à la limite inférieure de l'écran.

– Dans le cas d'un texte

Lorsque la position du caractère suivant dépasse la limite de l'écran, tout le contenu de l'écran défile de manière que le caractère suivant puisse être affiché dans les limites de l'écran. Le défilement se fait perpendiculairement au trajet-caractères.

b5 b4 Mode d'écriture par superposition

0 0 Les nouvelles données pour le plan de modèle remplacent les anciennes.

$M_i \leftarrow D_i$  (Mode de remplacement)

0 1 Les nouvelles données et les anciennes sont comparées selon «OU» et le résultat est écrit en mémoire.

$M_i \leftarrow M_i \text{ OU } D_i$

1 0 Les nouvelles données et les anciennes sont comparées selon ET et le résultat est écrit en mémoire.

$M_i \leftarrow M_i \text{ ET } D_i$

1 1 Les nouvelles données et les anciennes sont comparées selon OU exclusif et le résultat est écrit en mémoire.

$M_i \leftarrow M_i \text{ OU exclusif } D_i$

où  $M_i$  est le contenu de la mémoire du plan de modèle, et  $D_i$  est une donnée nouvelle.

Le mode de défaut est le mode remplacement.

### 3.3.2.7 TEXTE P

L'instruction TEXTE P est utilisée pour modifier les paramètres qui décrivent la manière d'affichage du texte. Le code d'opération est suivi d'un opérande à format fixe de deux octets et d'un opérande à valeur multiple (voir la figure 3.11).

L'opérande à format fixe détermine les paramètres de la manière suivante:

Espacement entre caractères. Les bits b5 et b6 de l'octet 1 sont utilisés pour déterminer la distance de déplacement du curseur (en multiples de dimension de champ de caractère parallèle au trajet-caractères) après l'affichage d'un caractère ou après la réception d'un caractère SPACE ou APB (retour) ou APF (tabulation horizontale).

b6 b5 Espacement entre caractères

0 0 1 (Défaut)

0 1 1,25

1 0 1,5

1 1 Proportionnel à la largeur du caractère

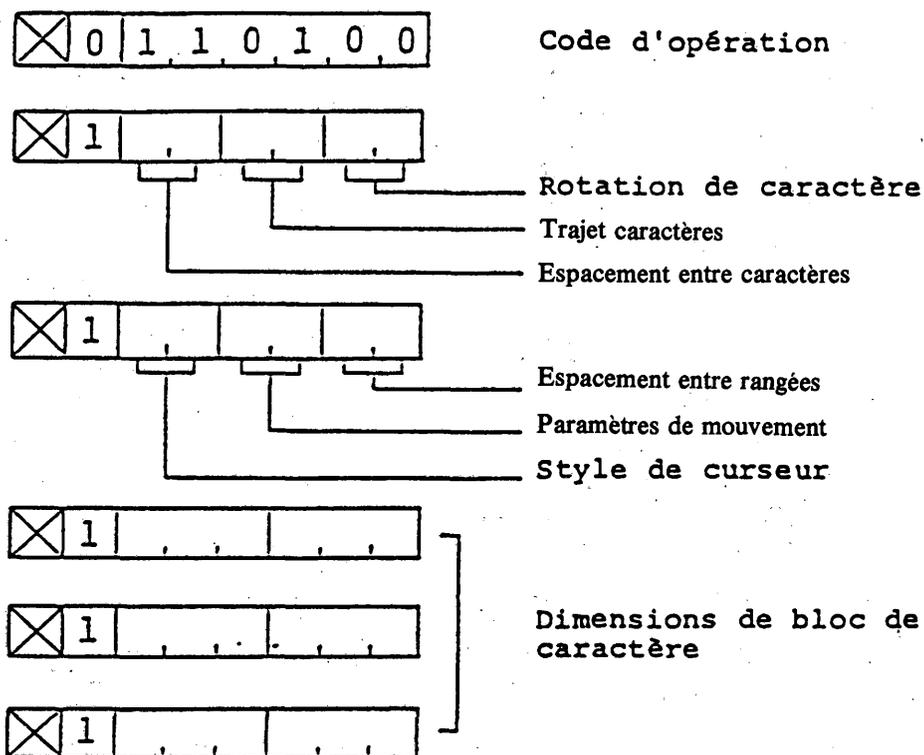


FIGURE 3.11

TEXTE P

Les trois espacements entre caractères fixes (1, 1,25 et 1,5) sont interprétés comme des fonctions multiplicatives de la dimension du champ de caractère actuel parallèle au trajet-caractères qui sont appliquées aux mouvements du curseur. Dans le mode d'espacement proportionnel, l'espacement entre caractères est une variable qui peut être fonction de la largeur du modèle déposé réel ainsi que de la taille du caractère actuelle et du style de la police. L'algorithme de l'espacement proportionnel dépend de l'utilisation. Cependant, chaque caractère doit être entièrement contenu dans la zone définie par le champ de caractère actuel. Cela signifie que le nombre exact de caractères par ligne n'est pas connu dans le mode d'espacement proportionnel, mais il est d'au moins autant de caractères par ligne que le permettent les dimensions du champ de caractère actuel. L'espacement entre caractères de défaut est un espace fixe de 1.

Mouvement du trajet-caractères. Les bits b3 et b4 de l'octet 1 déterminent la direction du trajet-caractères, c'est-à-dire la direction dans laquelle le curseur est automatiquement avancé après le dépôt d'un caractère. Le trajet-caractères est défini par rapport à l'horizontale à l'intérieur de l'écran unitaire et est indépendant de la rotation des caractères. Le trajet-caractères de défaut est vers la droite.

b4	b3	Mouvement du trajet-caractères
0	0	Droite (défaut)
0	1	Gauche
1	0	Haut
1	1	Bas

Rotation de caractères. Les bits b1 et b2 de l'octet 1 sont utilisés pour spécifier la rotation des caractères.

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

La rotation fait tourner le champ de caractère dans le sens inverse des aiguilles d'une montre par rapport à l'origine du champ de caractère. Cette rotation est mesurée par rapport à l'horizontale à l'intérieur de l'écran unitaire et est indépendante du trajet-caractères. L'origine du champ de caractère est le coin inférieur gauche du champ de caractère à la rotation de défaut de 0°, sans tenir compte du signe des dimensions de champ de caractère dx et dy. Toutes les images à l'intérieur d'un champ de caractère sont affectées par la rotation, de manière que la position relative de l'image à l'intérieur du champ de caractère soit inchangée.

Styles de curseur. Les bits b5 et b6 de l'octet 2 sont utilisés pour déterminer le style d'affichage du symbole de curseur.

b6	b5	Style de curseur
0	0	Souligné (défaut)
0	1	Bloc
1	0	Croix
1	1	Personnalisé

Ce curseur se trouve à la position dans laquelle le prochain caractère doit être déposé. Le symbole de curseur «souligné» est une ligne unique de la largeur du champ de caractère actuel, au bas du champ de caractère. Le symbole de curseur «bloc» est un bloc solide ou le contour d'un bloc dont la taille correspond à la taille du champ de caractère actuel. Le symbole de curseur «croix» consiste en une ligne verticale et une ligne horizontale qui se croisent au centre du champ de caractère et dont la hauteur et la largeur sont égales à la hauteur et à la largeur du champ de caractère actuel. La définition de la forme du symbole de curseur «personnalisé» est indépendante de l'utilisation.

Attributs de mouvement. Les bits b3 et b4 de l'octet 2 sont utilisés pour définir la relation entre le mouvement du curseur et le mouvement du point de dessin graphique.

b4	b3	Attributs de mouvement
0	0	Le curseur et le point de dessin se déplacent ensemble
0	1	Le curseur est en tête
1	0	Le point de dessin est en tête
1	1	Le curseur et le point de dessin se déplacent indépendamment (défaut)

Si le curseur et le point de dessin sont réglés pour se déplacer ensemble (00), lorsque le curseur se déplace (par exemple, des caractères sont affichés), le point de dessin graphique se déplace avec lui, en maintenant son alignement par rapport au curseur. De la même manière, lorsque le point de dessin se déplace (comme avec une primitive de dessin géométrique), le curseur se déplace également de manière à maintenir son alignement par rapport au point de dessin.

Lorsque le curseur est défini comme étant en tête (01), à chaque mouvement du curseur, le point de dessin se déplacera avec le curseur, mais pas vice versa.

Lorsque le point de dessin est mis en position «en tête» (10), à chaque mouvement du point de dessin, le curseur se déplacera avec celui-ci, mais pas vice versa.

Lorsque le point de dessin et le curseur sont réglés pour se déplacer indépendamment (11), le mouvement de l'un n'affectera en rien la position de l'autre.

L'alignement du point de dessin correspond à l'origine du champ de caractère pour le curseur «souligné» et pour le curseur «bloc», et au centre du champ de caractère pour le curseur «croix» et le curseur «personnalisé».

L'exécution d'une instruction TEXTE P entraînera l'alignement du point de dessin si l'attribut «mouvement ensemble» ou «curseur en tête» est en oeuvre après l'exécution. L'exécution d'une instruction TEXTE P n'aura pas d'effet sur la position de l'origine du champ de caractère.

Espacement entre rangées. Les bits b1 et b2 de l'octet 2 déterminent l'espacement entre rangées des caractères, qui définit l'emplacement relatif du curseur lorsque ce dernier avance sur une nouvelle ligne perpendiculairement ( $-90^\circ$ ) au trajet-caractères, soit automatiquement comme décrit ci-dessous, soit à l'aide des caractères APD (changement de ligne) ou APU (tabulation verticale). Les espacements entre rangées sont interprétés comme des fonctions multiplicatives de la dimension du champ de caractère perpendiculaire au trajet-caractères.

b2	b1	1 Espacement entre rangées
0	0	1 (défaut)
0	1	1,25
1	0	1,5
1	1	2

Lors de l'utilisation d'un espacement entre caractères fixe ou proportionnel, si l'origine du champ de caractère avance le long du trajet-caractères de manière qu'une quelconque partie du champ de caractère entier dépasse la limite de l'écran unitaire, un APR (retour chariot) automatique et un APD (changement de ligne) sont exécutés. Un espacement entre rangées de 1, dans lequel le champ de caractère de la rangée actuelle bute contre le champ de caractère de la rangée précédente correspond au défaut.

Dimensions de bloc de caractère. L'opérande à valeur multiple spécifie la largeur (dx) et la hauteur (dy) du bloc de caractère qui définit la taille de caractère «normale». (Les valeurs de défaut sont  $dx = 0,0625$  et  $dy = 0,09375$ ). La taille de bloc de caractère spécifiée reste inchangée jusqu'à ce qu'une autre instruction TEXTE P, NSR ou CS soit reçue.

Si l'opérande à valeur multiple est omis, la taille de bloc de caractère reste inchangée. Si un nombre supérieur d'opérandes suivent les codes d'opération spécifiés, ils sont rejetés.

Si dx est négatif, les modèles d'élément d'image sont réfléchis autour de l'axe central vertical du champ de caractère. Si dy est négatif, les modèles d'élément d'image sont réfléchis autour de l'axe central horizontal du champ de caractère.

### 3.3.2.8 ATTENTE P

L'instruction ATTENTE P entraîne un délai spécifique dans le traitement du texte et les instructions suivantes. Le format de l'instruction est indiqué sur la figure 3.12. Le code d'opération d'ATTENTE P est accompagné d'un opérande à valeur unique qui détermine la durée du délai en dixièmes de seconde.

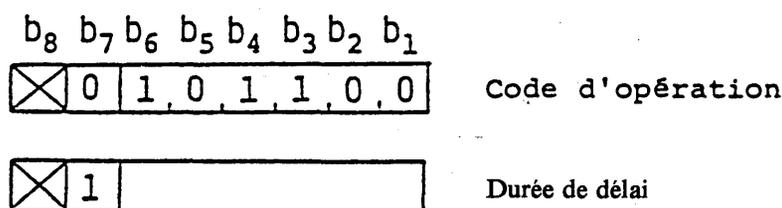


FIGURE 3.12  
ATTENTE P

### 3.3.3 Fonctions de commande multitrame

#### 3.3.3.1 Généralités

La présente norme comporte les fonctions permettant un affichage superposé de plusieurs trames et la partition de l'écran. La structure d'affichage multitrame permet l'affichage d'information sous forme de texte ou l'affichage d'information sous forme d'image animée sur des informations sous forme d'image d'arrière-plan.

La partition de l'écran permet différentes structures d'affichage: différentes structures multitrame et/ou compositions de trame peuvent être réalisées pour différentes portions de l'écran. Cette fonction permet l'affichage de différents types d'informations telles que des informations photographiques ou des informations géométriques dans un écran unique avec un minimum de mémoire de trame. Par exemple, la zone A de la figure 3.13 possède deux trames photographiques. Chaque trame consiste en M bits par pixel, alors que le reste de l'écran (zone B) possède trois trames de N bits. Si  $2M$  est égal à  $3N$ , les deux zones A et B peuvent entrer dans la mémoire de trame.

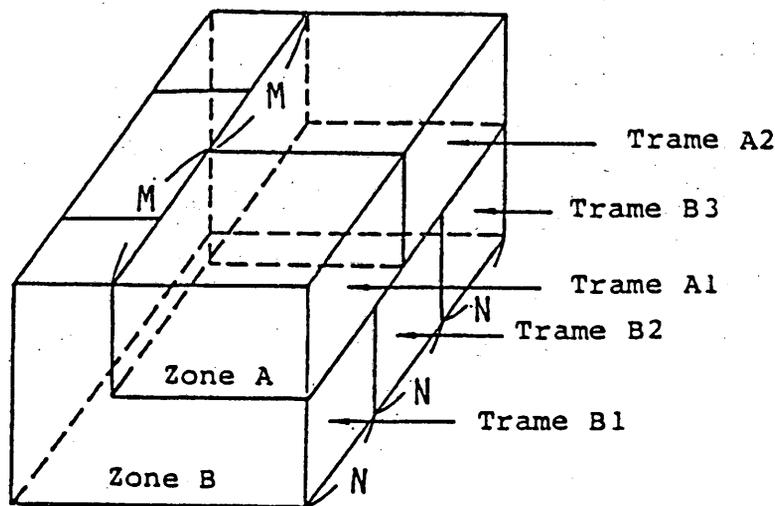


FIGURE 3.13

Notion de partition de l'écran

#### 3.3.3.2 Flux général des opérations d'affichage multitrame

Le flux général des opérations d'affichage multitrame est indiqué dans la figure 3.14.

Une zone de dessin active est d'abord définie. Les opérations suivantes, telles que l'affectation des bits, la sélection des trames et les instructions de dessin, ne sont effectives que dans la zone définie. On spécifie ensuite une structure multitrame et une affectation de bits pour chaque trame. Enfin, l'on choisit une trame désirée. Après cela, les instructions de dessin générales suivent.

#### 3.3.3.3 ZONE

L'instruction ZONE définit une zone de dessin active. Les instructions de dessin suivantes ne sont effectives que dans la zone définie. Le reste de l'écran est inchangé. Il est possible, par exemple, d'effectuer le défilement d'une image à l'intérieur d'une zone. Il n'y a qu'une zone de dessin active sur l'écran à la fois. Lors de la réception d'une nouvelle instruction ZONE, la définition de la zone est annulée.

Une zone est rectangulaire et définie par la position d'origine (x, y), la largeur (dX) et la hauteur (dY) (voir la figure 3.15). Le format de l'instruction est indiqué sur la figure 3.16. L'instruction ZONE proprement dite n'a pas d'effet sur l'image affichée.

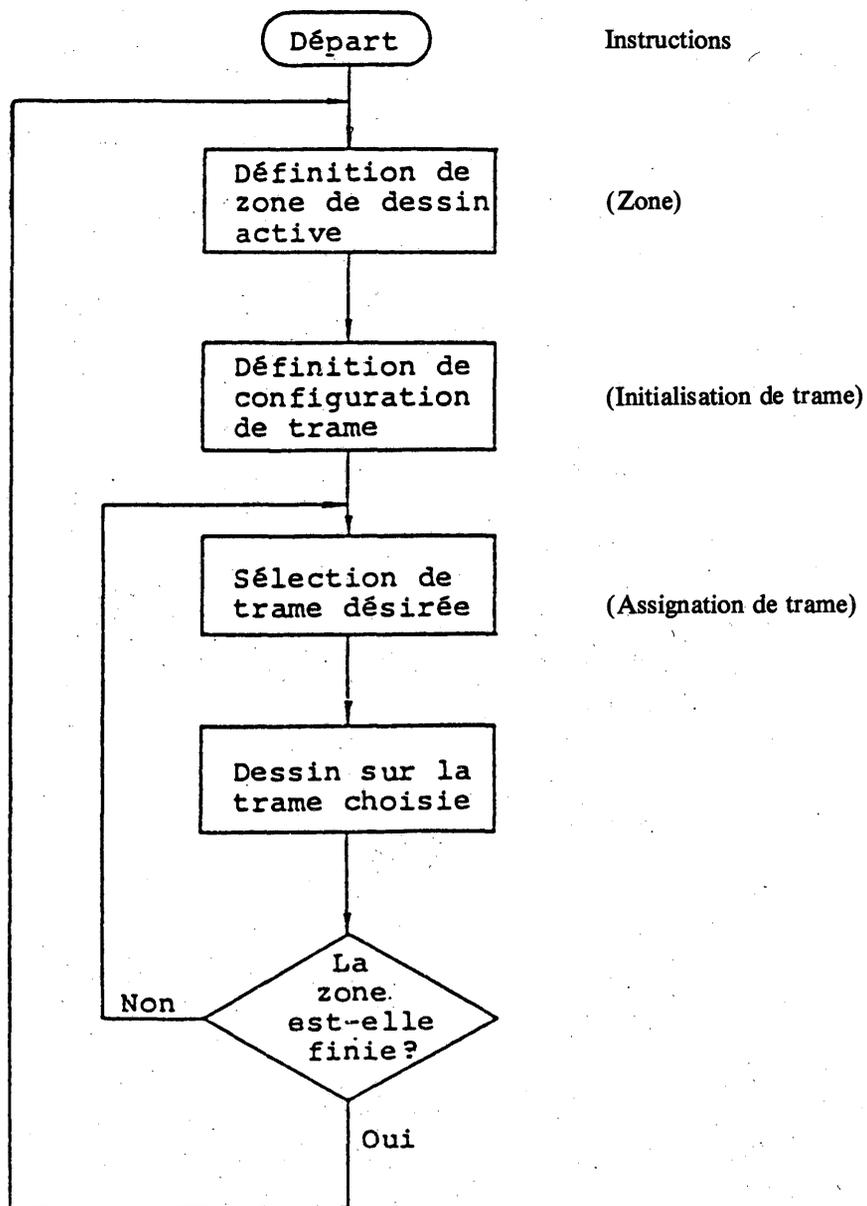


FIGURE 3.14

Flux général des opérations d'affichage multitrame

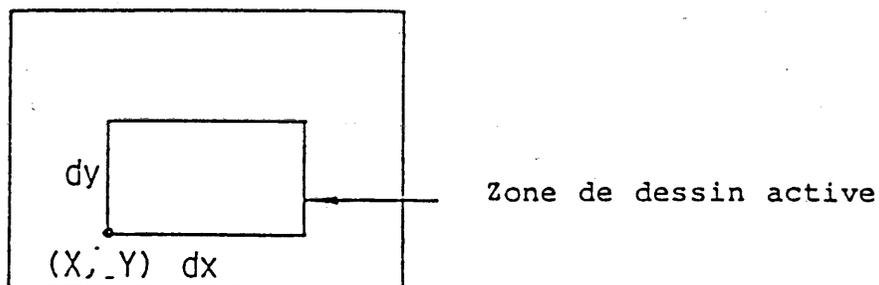


FIGURE 3.15

Définition de zone de dessin active

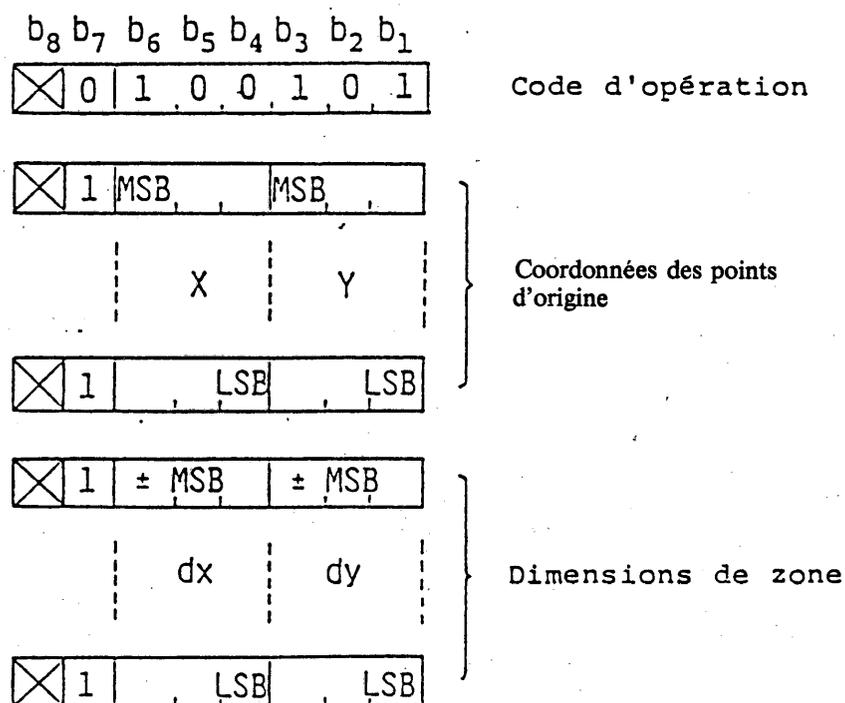


FIGURE 3.16

ZONE

### 3.3.3.4 INITIALISATION DE TRAME

L'instruction INITIALISATION DE TRAME spécifie la configuration de mémoire de trame relative à la zone définie. Le format est

Code d'opération a1, b1, a2, b2, ...,

comme indiqué sur la figure 3.17. Ici, «ai» est l'index de trame et «bi» est le nombre de bits par mémoires de trame attribuées à la trame. L'index de trame indique la priorité d'affichage relative. Si «ai» est inférieur à «aj», la trame «ai» a une priorité d'affichage supérieure à la trame «aj». Toute trame, cependant, a une priorité d'affichage supérieure à la couche d'arrière-plan de l'écran entier.

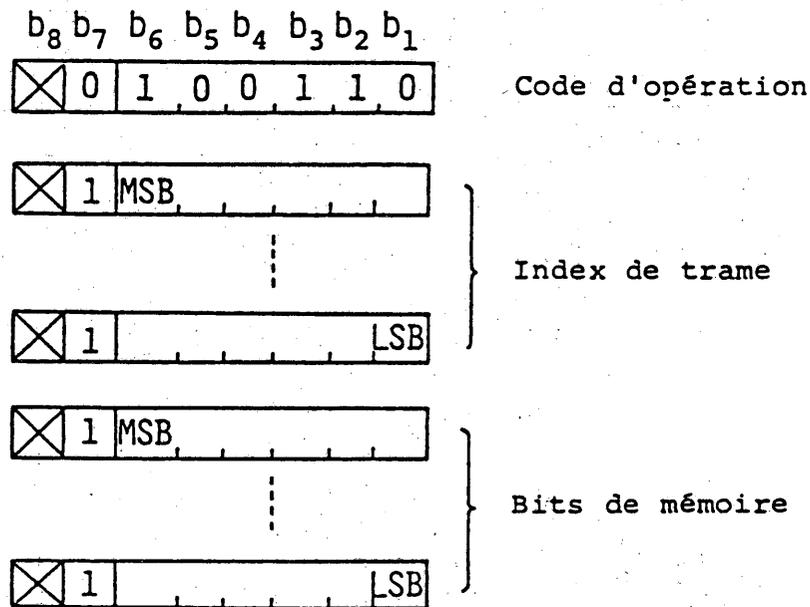


FIGURE 3.17  
INITIALISATION DE TRAME

### 3.3.3.5 ASSIGNATION DE TRAME

L'instruction ASSIGNATION DE TRAME permet la sélection de la trame sur laquelle sont effectués des dessins réussis. La trame est désignée par l'index de trame stipulé par l'instruction INITIALISATION DE TRAME. Le format de l'instruction est

Code d'opération ai,

comme indiqué sur la figure 3.18. Ici, «ai» est l'index de trame.

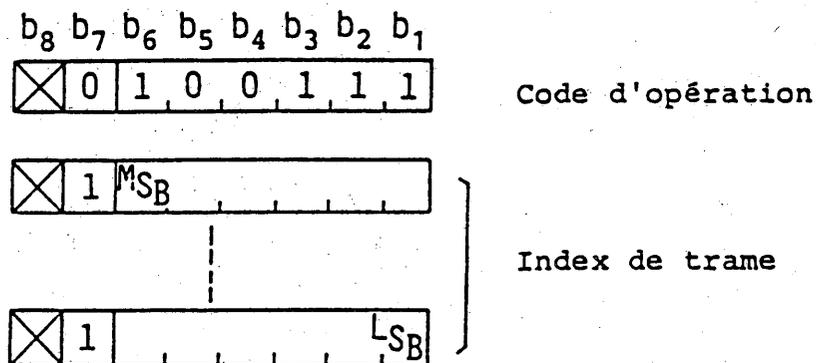


FIGURE 3.18  
ASSIGNATION DE TRAME

Le tableau 3.6 indique les instructions commandées par l'instruction ASSIGNATION DE TRAME. Ces instructions n'ont d'effet que dans la trame assignée.

TABLEAU 3.6

## Instructions commandées par ASSIGNATION DE TRAME

Jeu de commandes C0	Aucune
Jeu de commandes C1	Aucune
Jeu d'instructions de commande d'affichage	RÉINITIALISATION P, INITIALISATION LUT, CLIGNOTEMENT P et ATTENTE P *
Jeux de caractères	Toutes
Jeux de mosaïques	Toutes
Jeu de caractères dynamiquement redéfinissables	Toutes
Ensemble PDI	Toutes sauf RÉINITIALISATION, DOMAINE, TEXTURE, INITIALISATION COULEUR, ATTENTE, CHOIX COULEUR et CLIGNOTEMENT
Ensemble MVI	MOUVEMENT
Instructions photographiques	Toutes sauf JCDR P 1 et JCDR P 2

\* Applicable uniquement quand cette instruction est activée entre DÉBUT MVI et FIN MVI.

### 3.4 Jeux de caractères

#### 3.4.1 Jeu de caractères Kanji-japonais

Le jeu de caractères Kanji-japonais définit le total des 3657 caractères comprenant les caractères Kanji-japonais, les signes phonétiques japonais (Katakana et Hiragana), les caractères romains, des caractères numériques et des caractères supplémentaires. Le tableau 3.7 représente le jeu de caractères Kanji-japonais. Des codes à deux octets sont utilisés pour le jeu de caractères Kanji-japonais: le premier représente le numéro «Ku» et le deuxième le numéro «Ten». Les caractères 1 «Ku», 13 «Ten» à 18 «Ten» et 2 «Ku» 94 «Ten» sont définis comme caractères sans espacement. L'interprétation exacte de «Ku» ou de «Ten» est le numéro spécifié par l'octet plus 32.

Il est possible d'afficher les caractères Kanji-japonais en taille normale, largeur double, hauteur double et taille double.

#### 3.4.2 Jeu de caractères primaires

Le jeu de caractères primaires définit 52 caractères romains, 10 numériques et 32 marques. Le tableau 3.8 représente le jeu de caractères primaires. Un caractère primaire est représenté par un code à un seul octet. Toutes les tailles de caractère, y compris la taille moyenne et la petite taille sont valides pour les caractères primaires.

#### 3.4.3 Jeu de caractères Katakana

Le jeu de caractères Katakana définit 63 signes phonétiques japonais. Le tableau 3.9 représente le jeu de caractères Katakana. Un caractère Katakana est représenté par un code à un seul octet. Toutes les tailles de caractère sont valides.

### 3.5 Jeu de mosaïques

Deux jeux de mosaïques (jeu 1 et jeu 2) sont définis. Des codes à un seul octet sont utilisés pour les deux jeux.

Des formes contiguës et séparées sont prévues. Dans le mode soulignement activé par le caractère de commande C1 STL, tous les caractères des jeux de mosaïques sont affichés en représentation graphique séparée.

#### 3.5.1 Jeu de mosaïques 1

Le jeu de mosaïques 1 définit 59 caractères graphiques représentés dans le tableau 3.10.

#### 3.5.2 Jeu de mosaïques 2

Le jeu de mosaïques 2 définit 94 caractères graphiques qui sont compatibles avec le deuxième jeu supplémentaire de caractères mosaïques de syntaxe de données II (voir le tableau 3.11).









TABLEAU 3.8

Jeu de caractères primaires

					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	<del>0</del>	0	@	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	/	?	0	-	0	<del>0</del>	

TABLEAU 3.9

Jeu de caractères Katakana

					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	X					
0	0	1	0	2						
0	0	1	1	3						
0	1	0	0	4						
0	1	0	1	5					Réservé	Réservé
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						X

2(10)/7 ~ 2(10)/15: Petite taille

TABLEAU 3.10  
Jeu de mosaïques 1

					b7	0	0	0	0	1	1	1	1							
					b6	0	0	1	1	0	0	1	1							
					b5	0	1	0	1	0	1	0	1							
						0	1	2	3	4	5	6	7							
b4	b3	b2	b1																	
0	0	0	0	0																
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																

### 3.6 *Jeux de caractères dynamiquement redéfinissables (JCDR)*

Le jeu de caractères dynamiquement redéfinissables (JCDR) permet le chargement de modèles personnalisés définis et leur utilisation en tant que jeu G. Le chargement est exécuté dans le mode transparent par les instructions JCDR 1 ou 2.

#### 3.6.1 *JCDR 1*

Les modèles JCDR 1 sont identifiés par des codes à un seul octet. Un code JCDR possède une valeur dans la gamme de 2/1 à 7/14. Le chargement du modèle est exécuté par l'instruction JCDR photo 1.

#### 3.6.2 *JCDR 2*

Les modèles JCDR 2 sont identifiés par des codes à deux octets. Un code JCDR possède une valeur dans la gamme de 2/1 2/1 à 7/14 7/14. Le chargement du modèle est exécuté par l'instruction JCDR photo 2.

### 3.7 *Jeu de macros*

Une instruction macro consiste en une chaîne arbitraire de code de niveau de présentation stocké localement, identifiée par une position de code (nom de macro) à partir du jeu de macros G. Ce nom (de 2/0 à 7/15) agit ensuite comme substitut pour toute la chaîne de codes qui est chargée par l'instruction macro photo.

### 3.8 *Jeu d'instructions de description d'image (PDI)*

Le jeu PDI est identique à celui de la syntaxe de données III à l'exception de l'instruction TEXTE (voir le tableau 3.12). Cependant, les gabarits de texture programmables ne sont pas supportés.

Les instructions suivantes dans le jeu PDI et l'instruction de commande d'affichage ont des effets identiques sur les attributs d'affichage.

<i>Jeu PDI</i>	<i>Ensemble d'instructions de commande d'affichage</i>
RÉINITIALISATION	RÉINITIALISATION P
DOMAINE	DOMAINE P, PEL LOGIQUE
INITIALISATION COULEUR	INITIALISATION LUT
ATTENTE	ATTENTE P
CLIGNOTEMENT	CLIGNOTEMENT P

L'instruction SÉLECTION COULEUR a le même effet que le caractère de commande couleur C1.

En cas d'instructions divergentes, c'est la dernière reçue qui commande les attributs d'affichage. En d'autres mots, les attributs d'affichage sont les mêmes pour les instructions PDI et les instructions photographiques.

### 3.9 *Jeu d'instructions de mouvement (MVI)*

L'instruction MOUVEMENT (voir le § 3.9.5) entraîne le déplacement virtuel de la trame spécifiée par ASSIGNATION DE TRAME par rapport aux autres trames (voir la figure 3.19).

Il est possible, en utilisant cette instruction et la structure de trame multiple, d'obtenir la présentation d'images animées.

Une instruction de mouvement (MVI) se compose d'un code d'opération et d'opérandes similaires aux PDI (certaines instructions ne comprennent pas d'opérandes).

Il s'agit d'opérandes à valeur unique, d'opérandes à valeur multiple et d'opérandes à format fixe. La longueur en octet d'un opérande à valeur unique et celle d'un opérande à valeur multiple sont spécifiées par l'instruction DOMAINE P dans le jeu d'instructions de commande d'affichage.

Le jeu MVI est indiqué dans le tableau 3.13.

TABLEAU 3.11  
Jeu de mosaïques 2

					b7	0	0	0	0	1	1	1	1							
					b6	0	0	1	1	0	0	1	1							
					b5	0	1	0	1	0	1	0	1							
						0	1	2	3	4	5	6	7							
b4	b3	b2	b1																	
0	0	0	0	0																
1	0	0	0	8																

TABLEAU 3.12

Jeu PDI

					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	Réinitialisation	Rectangle (contour)	Données numériques				
0	0	0	1	1	Domaine	Rectangle (plein)					
0	0	1	0	2		Initialisation & Rect. (contour)					
0	0	1	1	3	Texture	Initialisation & Rect. (plein)					
0	1	0	0	4	Point Initialisation (ABS)	Polygone (contour)					
0	1	0	1	5	Point Initialisation (REL)	Polygone (plein)					
0	1	1	0	6	Point (ABS)	Initialisation & Poly. (contour)					
0	1	1	1	7	Point (REL)	Initialisation & Poly. (plein)					
1	0	0	0	8	Ligne (ABS)	Champ					
1	0	0	1	9	Ligne (REL)	Point d'incrément					
1	0	1	0	10	Initialisation & Ligne (ABS)	Ligne d'incrément					
1	0	1	1	11	Initialisation & Ligne (REL)	Poly. d'incrément (plein)					
1	1	0	0	12	ARC (contour)	Initialisation couleur					
1	1	0	1	13	ARC (plein)	Statut de commande (attente)					
1	1	1	0	14	Initialisation & ARC (contour)	Sélection couleur					
1	1	1	1	15	Initialisation & ARC (plein)	Clignotement					

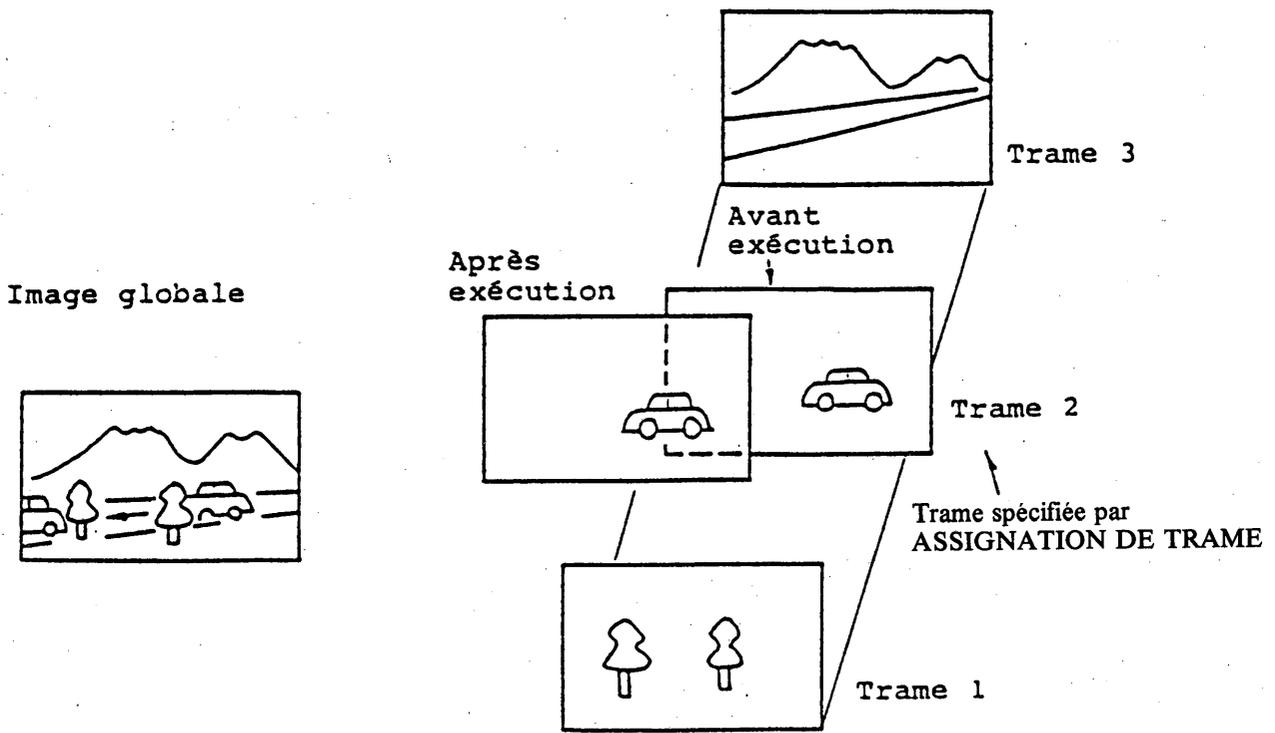


FIGURE 3.19

Exemples d'instructions de mouvement

3.9.1 DÉBUT MVI

DÉBUT MVI indique que tous les caractères entre DÉBUT MVI et FIN MVI doivent être enregistrés dans la mémoire tampon MVI (voir la figure 3.20). DÉBUT MVI n'a pas d'opérande.

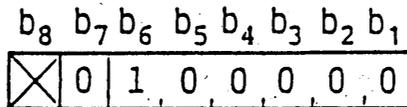


FIGURE 3.20

DÉBUT MVI

3.9.2 FIN MVI

FIN MVI met fin aux instructions d'enregistrement puis active l'exécution des instructions enregistrées (voir la figure 3.21). FIN MVI n'a pas d'opérande. Lorsque tous les caractères enregistrés sont exécutés, la mémoire tampon MVI est vidée.

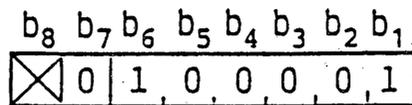


FIGURE 3.21

FIN MVI

TABLEAU 3.13

Jeu MVI

					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	Début MVI		Données numériques				
0	0	0	1	1	Fin MVI						
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		Mouvement					
1	0	1	1	11							
1	1	0	0	12		Début Répétition					
1	1	0	1	13		Fin Répétition					
1	1	1	0	14							
1	1	1	1	15							

### 3.9.3 DÉBUT RÉPÉTITION

DÉBUT RÉPÉTITION indique que tous les caractères de niveau présentation entre DÉBUT RÉPÉTITION et FIN RÉPÉTITION sont exécutés de manière répétée. Le nombre de répétitions est spécifié par l'opérande DÉBUT RÉPÉTITION (voir la figure 3.22).

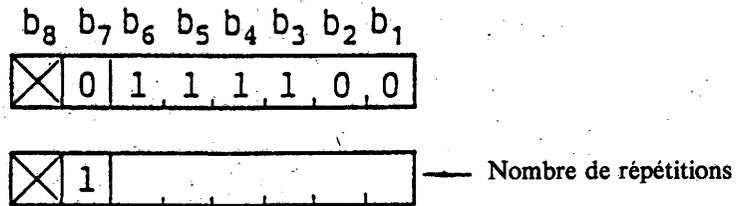


FIGURE 3.22  
DÉBUT RÉPÉTITION

### 3.9.4 FIN RÉPÉTITION

FIN RÉPÉTITION termine la répétition des chaînes de caractères FIN RÉPÉTITION n'a pas d'opérande (voir la figure 3.23).

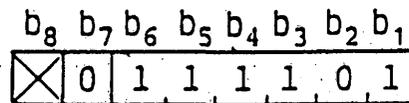


FIGURE 3.23  
FIN RÉPÉTITION

### 3.9.5 MOUVEMENT

MOUVEMENT prend deux opérandes à valeur multiple, suivis d'un opérande à valeur unique (voir la figure 3.24). Le premier opérande à valeur multiple spécifie les coordonnées du point d'origine sur la trame désignée par ASSIGNATION de TRAME. Le deuxième opérande à valeur multiple spécifie les coordonnées du point de terminaison. La trame spécifiée se déplace du point d'origine au point de terminaison. L'opérande à valeur unique suivant immédiatement le deuxième opérande à valeur multiple indique la période de temps (en dixièmes de seconde) pendant laquelle la trame se déplace du point d'origine au point de terminaison.

Les coordonnées du point d'origine représentent les coordonnées du coin inférieur gauche de la trame spécifiée par ASSIGNATION de TRAME par rapport à l'écran unitaire. Les coordonnées du point de terminaison sont spécifiées en coordonnées relatives par rapport à celles du point d'origine. Pour la description du mouvement, les limites de l'écran unitaire sont étendues à  $-1,0$  pour les directions X et Y.

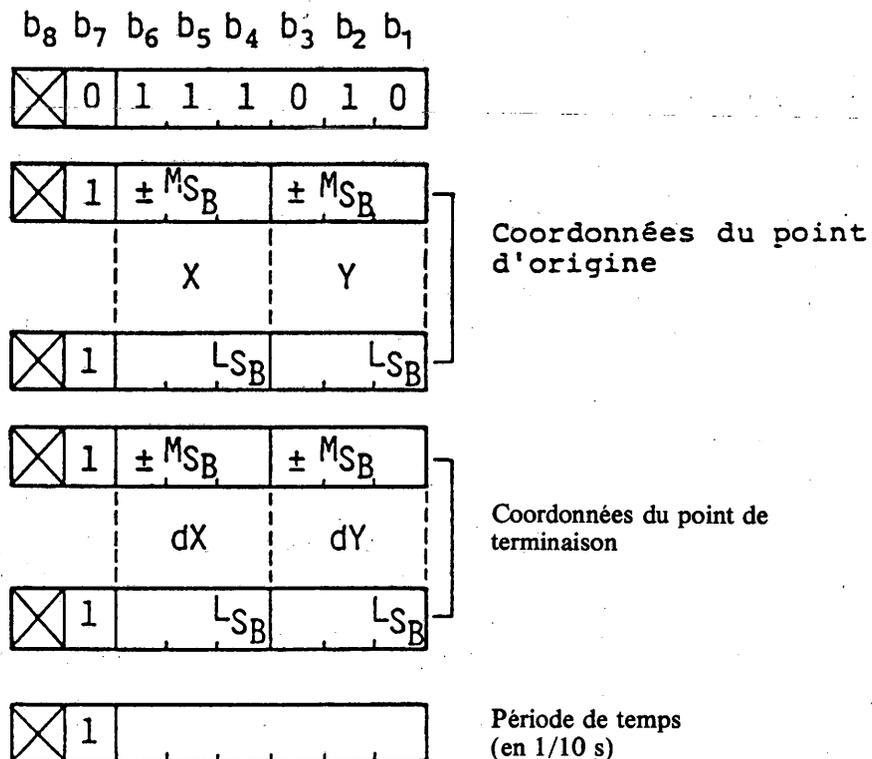


FIGURE 3.24  
MOUVEMENT

#### 4 Codage en mode transparent

##### 4.1 Unité de données photographiques (PDU)

Une unité de données photographiques (PDU) se compose d'un code d'opération avec un indicateur de longueur (LI), suivi par zéro, un ou plus opérandes chacun comprenant un ou plus octets de séquences de bits.

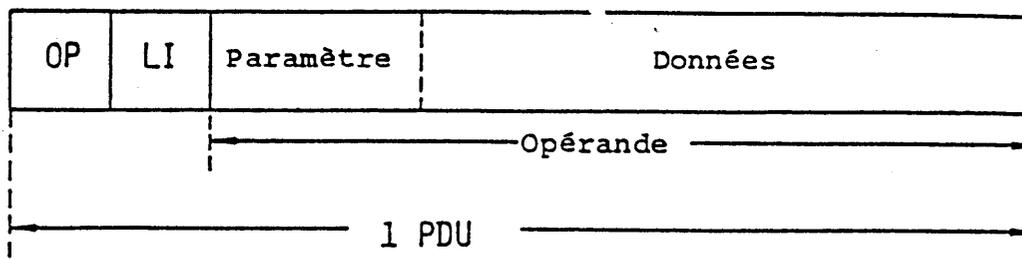
L'utilisation de toutes les formes d'octet est permise dans les séquences de bits des opérandes, résultant en une expression efficace de données arbitraires. Le format de PDU est indiqué dans la figure 4.1.

Le code d'opération consiste en un octet unique qui indique le sens des données photographiques contenues dans la PDU.

Le LI consiste en un ou plus octets. La valeur d'un LI est un nombre binaire qui représente la longueur totale des opérandes suivant le champ de LI dans l'octet. Si l'expression décimale du deuxième octet est 255, les deux octets suivants (le troisième et le quatrième) indiquent la longueur réelle. Comme le LI indique simplement la longueur des données, il n'est pas indiqué dans la figure du format de PDU dans le présent paragraphe.

Un ou plus octets de paramètres sont situés dans la partie d'attaque d'un champ d'opérande. Les paramètres comprennent les coordonnées de point de dessin où les données photographiques devraient être affichées et/ou le format de compactage qui indique la manière dont les données photographiques sont disposées. Les données photographiques exprimées point par point sont contenues dans la partie restante du champ d'opérande.

Ici, le point correspond au pel logique ou au bloc de coloration dans l'écran unitaire. La position ou les coordonnées du pel sont définies comme l'une des quatre positions de coin de l'élément d'image logique, en fonction des signes de la largeur ( $dx_0$ ) et de la hauteur ( $dy_0$ ) du pel logique. La position du bloc de coloration est définie de la même manière que la position du pel, pour autant que le terme «pel logique» soit remplacé par «bloc de coloration».



OP: Code d'opération  
 LI: Indicateur de longueur

FIGURE 4.1  
 Format de la PDU

Il existe trois types d'opérands de paramètre: format fixe, valeur unique et valeur multiple. Les longueurs des opérands de paramètre à valeur unique ou à valeur multiple sont déterminées par l'instruction DOMAINE P.

Les codes d'opération de la PDU sont indiqués dans le tableau 4.1. Les blancs dans le tableau sont réservés aux codes d'opération pour la coloration point par point (c'est-à-dire MIC, MICDA et codage par transformation), audio, télélogiciel, etc.

#### 4.2 CONFIGURATION DE POINTS EN LIGNE

Le format de la PDU CONFIGURATION DE POINTS EN LIGNE est indiqué sur la figure 4.2. Les données photographiques expriment les valeurs des pels logiques, 1 et 0, qui doivent être inscrits le long d'une ligne horizontale sur le plan de modèle.

Un opérande de paramètre à valeur unique indique les coordonnées absolues où les données doivent être écrites.

La correspondance entre les données photographiques et les positions des pels logiques dans le plan de modèle est indiquées sur la figure 4.3.

Le premier bit  $P_0$  est déposé à la position du pel logique la plus à gauche, le bit  $P_1$  qui suit à la position de pel suivant à droite. Lorsque le bit suivant est reçu, la position du pel se déplace vers la droite sur une distance égale à la taille du pel. Si le bit  $P_n$  correspond au pel logique le plus à droite, la position du pel correspondant au bit  $P_{n+1}$  suivant devra être déplacé à la position de pel la plus à gauche, sur une distance égale à la hauteur du pel logique vers le bas.

Si un bit correspond à la position de pel du coin inférieur droit et qu'il reste encore des bits, ceux-ci sont rejetés.

#### 4.3 CONFIGURATION DE POINTS EN LIGNE AVEC COMPRESSION

Ce format de PDU est représenté sur la figure 4.4. Les données photographiques expriment les formes codées de longueur de passage des valeurs de pels logiques, qui doivent être inscrites de la même manière que le format CONFIGURATION.

L'opérande de paramètre à valeur unique indique la coordonnée absolue Y où les données sont écrites.

La compression des données est obtenue à l'aide de la technique de codage de longueur de passage Hoffman modifiée (MH). Le processus de codage commence avec le passage des 0, puis continue alternativement pour les passages des 1 et les passages des 0. Le dernier passage dans une ligne est codé comme code de fin de ligne (EOL), sans tenir compte de sa longueur de passage, car il va jusqu'à la fin de la ligne.

TABLEAU 4.1

Codes d'opération PDU

b <sub>8</sub>	0	0	0	0	0	0		
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	8 ~ 15
0 0 0 0	0	Configuration de points en ligne (avec compression)	Configuration de points en ligne	Bloc de coloration (avec compression)	Bloc de coloration	Télélogiciel		
0 0 0 1	1							
0 0 1 0	2		Configuration de points en champ	Bloc de coloration de champ (avec compression)	Bloc de coloration de champ			* Télélogiciel * Données audio
0 0 1 1	3							
0 1 0 0	4			Bloc de coloration de format libre				* Coloration point par point avec réduction de redondance
0 1 0 1	5							
0 1 1 0	6							* Coloration point par point sans réduction de redondance
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	JCDR 1 Photo						
1 1 0 1	13	JCDR 2 Photo						
1 1 1 0	14							
1 1 1 1	15							
								* En étude



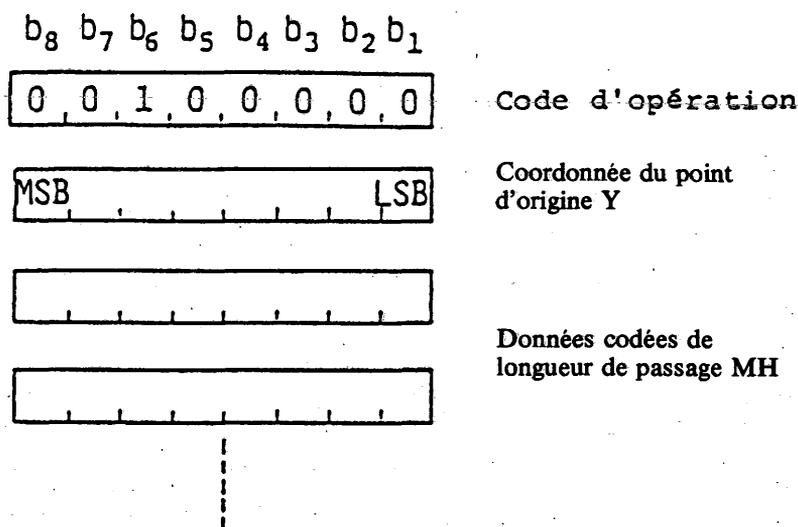


FIGURE 4.4  
**CONFIGURATION DE POINTS EN LIGNE AVEC COMPRESSION**

La séquence de bits suivant le EOL doit être décodée pour les pels logiques situés à une hauteur de pel logique au-dessous. Le processus de codage commence avec le passage des 0 et continue de la même manière pour la première ligne.

Le tableau de code MH fait l'objet des tableaux 4.2 et 4.3. Un exemple de codage est indiqué sur la figure 4.5.

Si plus de données sont reçues lorsque la position du pixel atteint la limite inférieure de l'écran, elles sont rejetées.

#### 4.4 CONFIGURATION DE POINTS EN CHAMP

Le format de la PDU CONFIGURATION DE POINTS EN CHAMP est représenté sur la figure 4.6. Les données photographiques expriment les valeurs des pels dans un champ rectangulaire sur le plan de modèle.

Il y a deux opérandes de paramètres à valeur multiple qui définissent l'emplacement et les dimensions du champ rectangulaire dans lequel les données photographiques doivent être déposées.

Le premier opérande à valeur multiple spécifie la position du pel logique initial en coordonnées absolues. La largeur et la hauteur (dx, dy) sont données par le deuxième opérande à valeur multiple. La position du pel initial peut être l'un des quatre sommets du champ car la largeur et la hauteur peuvent être positives ou négatives.

La correspondance entre les données et les pels logiques dans le plan de modèle est indiquée sur la figure 4.7. Après le dépôt d'un bit de la séquence de données photographiques à la grandeur du pel logique actuel, la position du pel est automatiquement déplacée dans la direction x d'une distance égale à la largeur (dx<sub>0</sub>) du pel logique. Le prochain bit est déposé à cette position. Si la dimension de champ dx est positive, la position se déplace vers la droite; si elle est négative, la position se déplace vers la gauche.

TABLEAU 4.2

**Table de code de longueur de passage MH  
(partie 1: passage des 0)**

	Code de terminaison		Code d'appoint
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	Réservé	000000xx---

TABLEAU 4.3

**Table de code de longueur de passage MH  
(partie 2: passage des 1)**

	Code de terminaison		Code d'appoint
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	Réservé	000000xx----

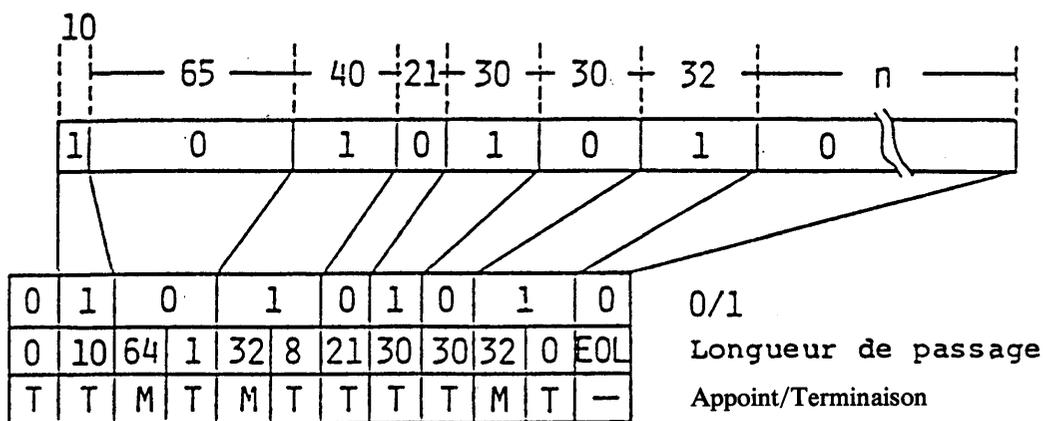


FIGURE 4.5

Un exemple de codage de CONFIGURATION DE POINTS EN LIGNE AVEC COMPRESSION

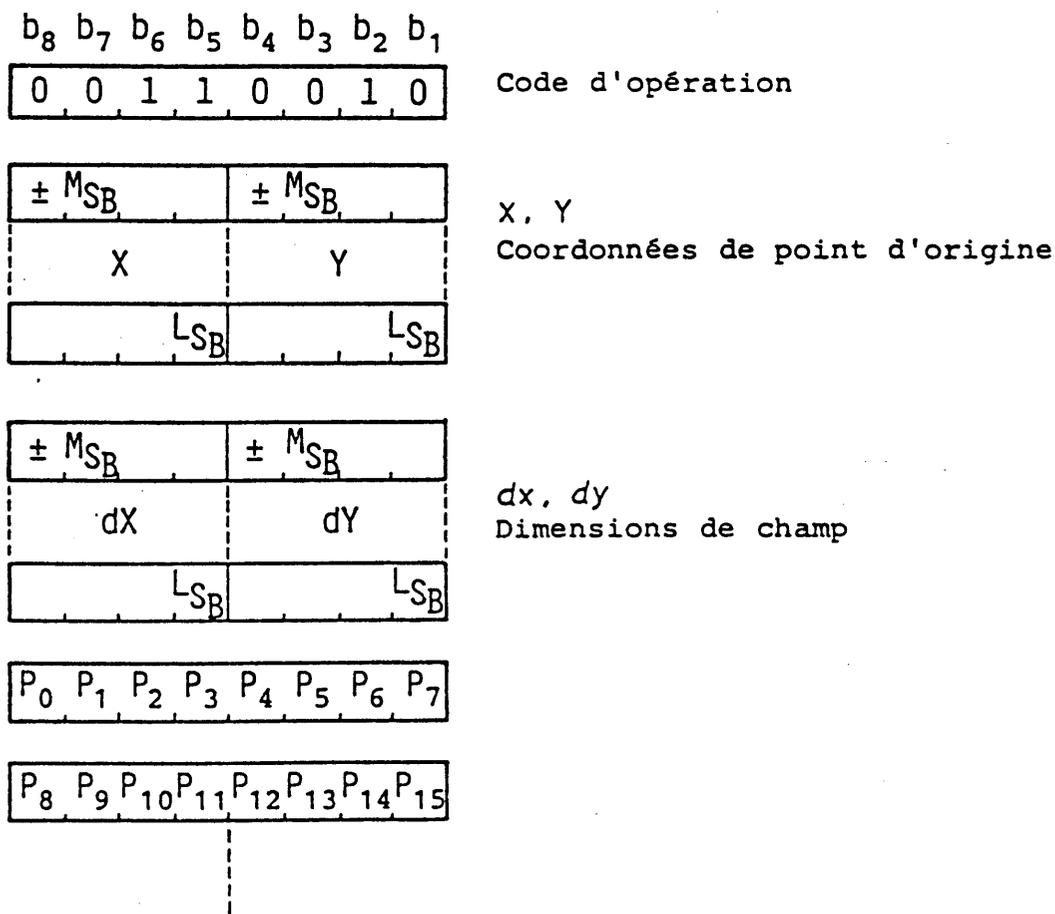


FIGURE 4.6

CONFIGURATION DE POINTS EN CHAMP

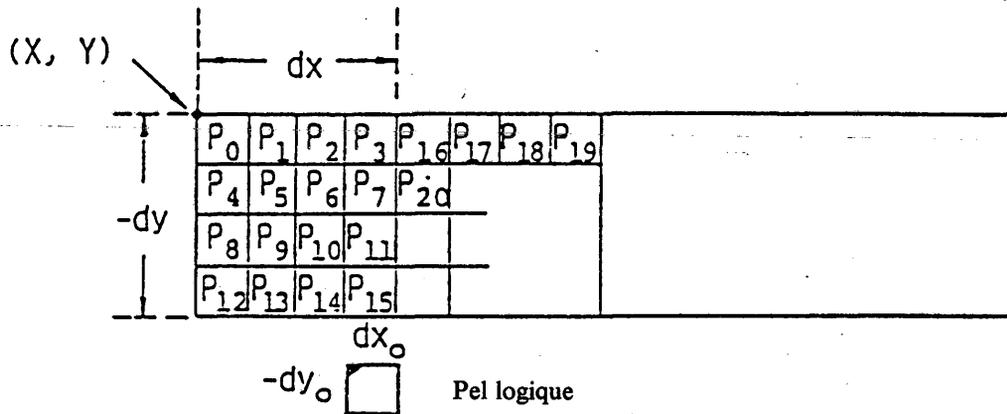


FIGURE 4.7

**Correspondance entre les données photographiques et les positions des pels logiques**

Lorsque la position du pel atteint ou dépasse un côté vertical dans le champ, la coordonnée  $x$  du pel est déplacée de l'autre côté du champ et la coordonnée  $y$  est déplacée sur une distance égale à la hauteur ( $dy_0$ ) du pel logique. Si la dimension du champ  $dy$  est positive, la position du pel se déplace vers le haut; si elle est négative, la position se déplace vers le bas.

Lorsque la position du pel atteint ou dépasse un côté horizontal du rectangle, et s'il reste des bits, ces derniers doivent être interprétés comme s'ils poursuivent le même code d'opération avec les opérandes de paramètre de champ  $(X + dx, Y)$ ,  $(dx, dy)$ .

**4.5 BLOC DE COLORATION**

Le format de la PDU BLOC DE COLORATION est indiqué sur la figure 4.8. Les données photographiques expriment la couleur et les attributs des blocs de coloration sur une ligne horizontale sur les plans de FG, de BG et d'attribut d'affichage.

Le premier opérande de paramètre de format fixe indique les drapeaux d'attribut et de couleur. Si le drapeau de la couleur FG, de la couleur BG ou de l'attribut d'affichage est «1», la PDU contient les données correspondantes.

Le deuxième paramètre, un opérande à valeur unique, indique la coordonnée  $Y$  absolue où commence le décodage des données d'attribut.

Les données photographiques se composent de trois groupes successifs de codes de bits qui représentent les données FG, les données BG et les attributs d'affichage. Un quatre-bits pour les données FG ou données BG est une adresse d'entrée du LUT. Les attributs d'affichage de quatre-bits sont utilisés pour les commandes de clignotement, et leur signification est indiquée dans le tableau 4.4. Lorsque le premier opérande indique que tous les drapeaux sont des 1, les données de couleur FG pour la première ligne viennent en premier, les données de couleur BG pour la même ligne viennent ensuite et les données d'attribut d'affichage pour la même ligne suivent. S'il reste encore des données, elles sont interprétées comme données pour les blocs de coloration dans une ligne qui se trouve une hauteur de bloc de coloration au-dessous. Les données restantes sont interprétées de la même manière que la première ligne.

Toute limite de données dans des octets d'opérande (entre les données de couleur FG et les données de couleur BG, entre les données d'attributs d'affichage et les données de couleur BG ainsi qu'entre une ligne et la ligne qui suit) maintient la limite des octets, c'est-à-dire si les dernières données n'occupent qu'une portion d'un octet, les bits restants de l'octet sont remplis de bits 0.

Si les données d'opérande dépassent la limite du bloc le plus bas, elles ne sont pas prises en compte.

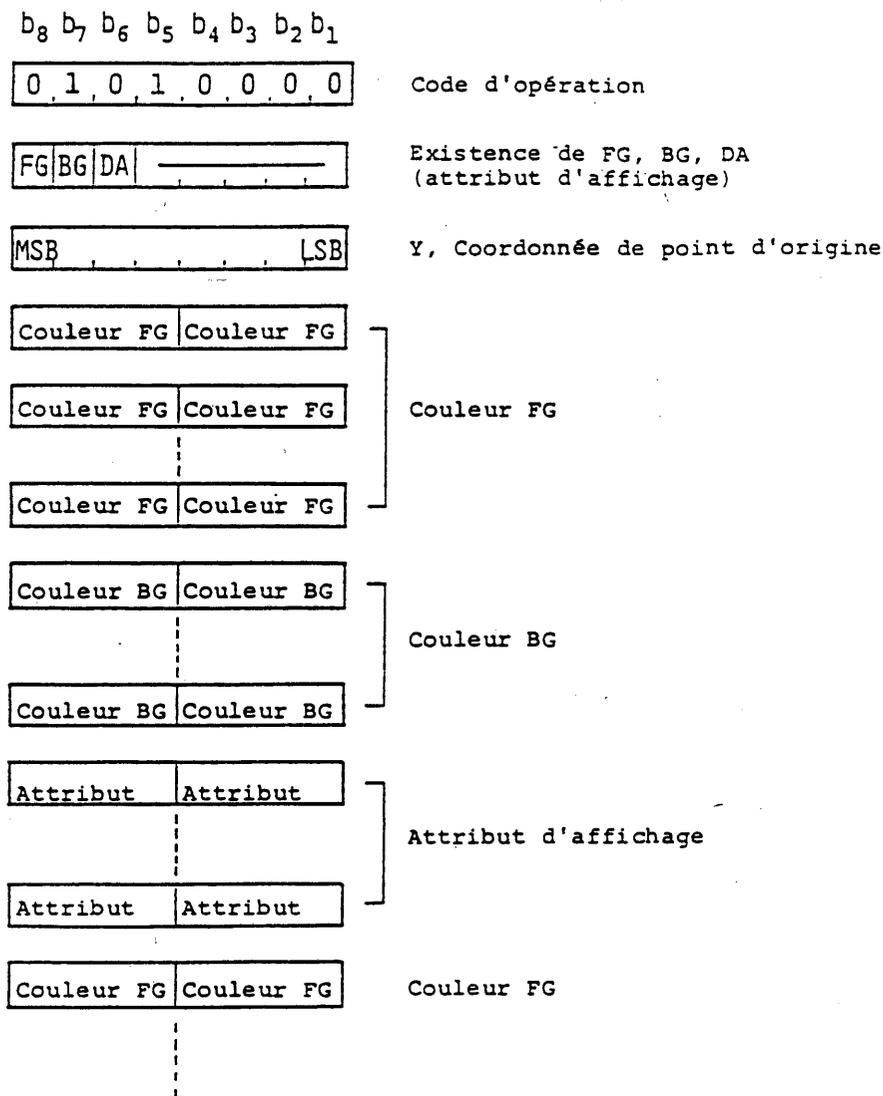


FIGURE 4.8  
BLOC DE COLORATION

#### 4.6 BLOC DE COLORATION AVEC COMPRESSION

Le format de la PDU BLOC DE COLORATION AVEC COMPRESSION est indiqué sur la figure 4.9. Les données photographiques expriment la couleur et l'attribut des blocs de coloration sur une ligne horizontale dans les plans de FG, de BG et d'attribut d'affichage.

Le premier opérande de paramètre de format fixe indique les drapeaux d'attribut. Si le drapeau de couleur FG, de couleur BG ou d'attribut d'affichage est «1», la PDU contient les données correspondantes.

Le deuxième paramètre, un opérande à valeur unique, indique la coordonnée Y absolue, où commence le décodage des données d'attribut.

Les données photographiques sont composées de paires de code attribut/passage.

Dans une paire attribut/passage, quatre bits sont affectés à l'attribut et au passage. Un quatre-bits pour les plans FG ou BG est une adresse d'entrée du LUT. Les attributs de quatre-bits pour les plans d'attribut d'affichage sont utilisés pour les commandes de clignotement, et leur signification est indiquée dans le tableau 4.4. Le tableau de code de longueur de passage quatre-bits fait l'objet du tableau 4.5. Le dernier passage sur une ligne horizontale est codé comme ECL (fin de ligne de coloration), quelle que soit sa longueur.

TABLEAU 4.4

Code d'attribut d'affichage

Code	Clignotement	Phase	Modèle
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	—	—

Si les drapeaux de l'opérande de format fixe sont tous des «1», les paires couleur/passage FG pour la première ligne viennent en premier, les paires couleur/passage BG viennent ensuite et les paires attribut d'affichage/passage pour la même ligne viennent ensuite. S'il reste des données, elles sont interprétées comme données pour les blocs de coloration sur une ligne située une hauteur de bloc de coloration au-dessous. Les données restantes sont interprétées de la même manière que la première ligne.

S'il y a un drapeau «0» dans l'opérande de format fixe, les données lui correspondant sont omises de l'expression.

Un exemple de codage est indiqué sur la figure 4.10.

4.7 BLOC DE COLORATION DE CHAMP

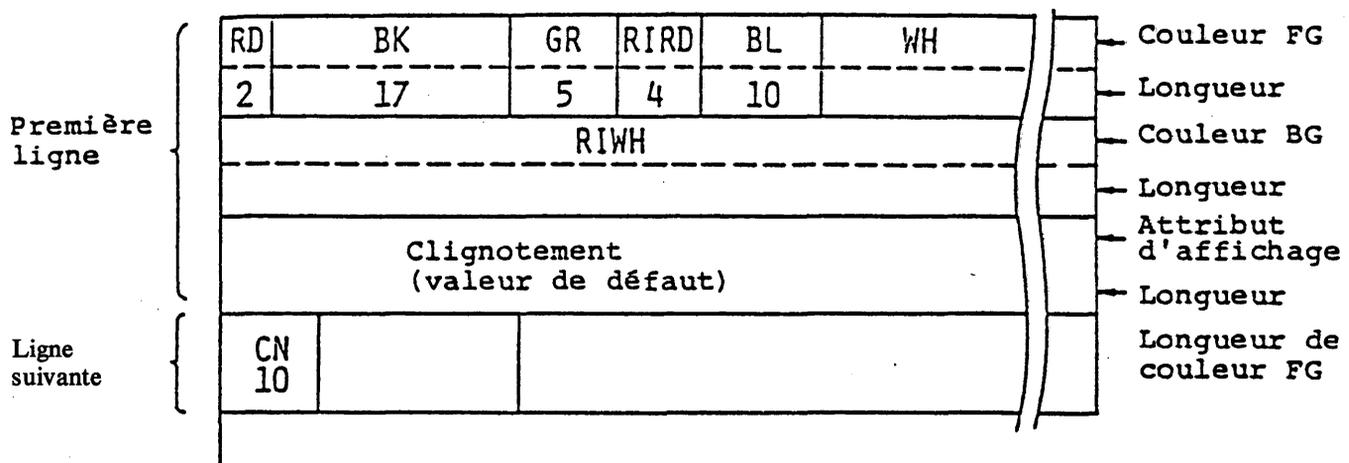
Le format de la PDU est indiqué sur la figure 4.11. Les données photographiques expriment les couleurs et les attributs d'affichage des blocs de coloration dans un champ sur les plans de FG, de BG et d'attribut d'affichage.

TABLEAU 4.5

Code de longueur de passage 4-bits

Longueur de passage	Code	Longueur de passage	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	Couleur FG
			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	Couleur BG
			RIWH	0	0	0	0	ECL	
			CN	1	0	1	0	10	Couleur FG

FIGURE 4.10

Exemple de codage BLOC DE COLORATION AVEC COMPRESSION

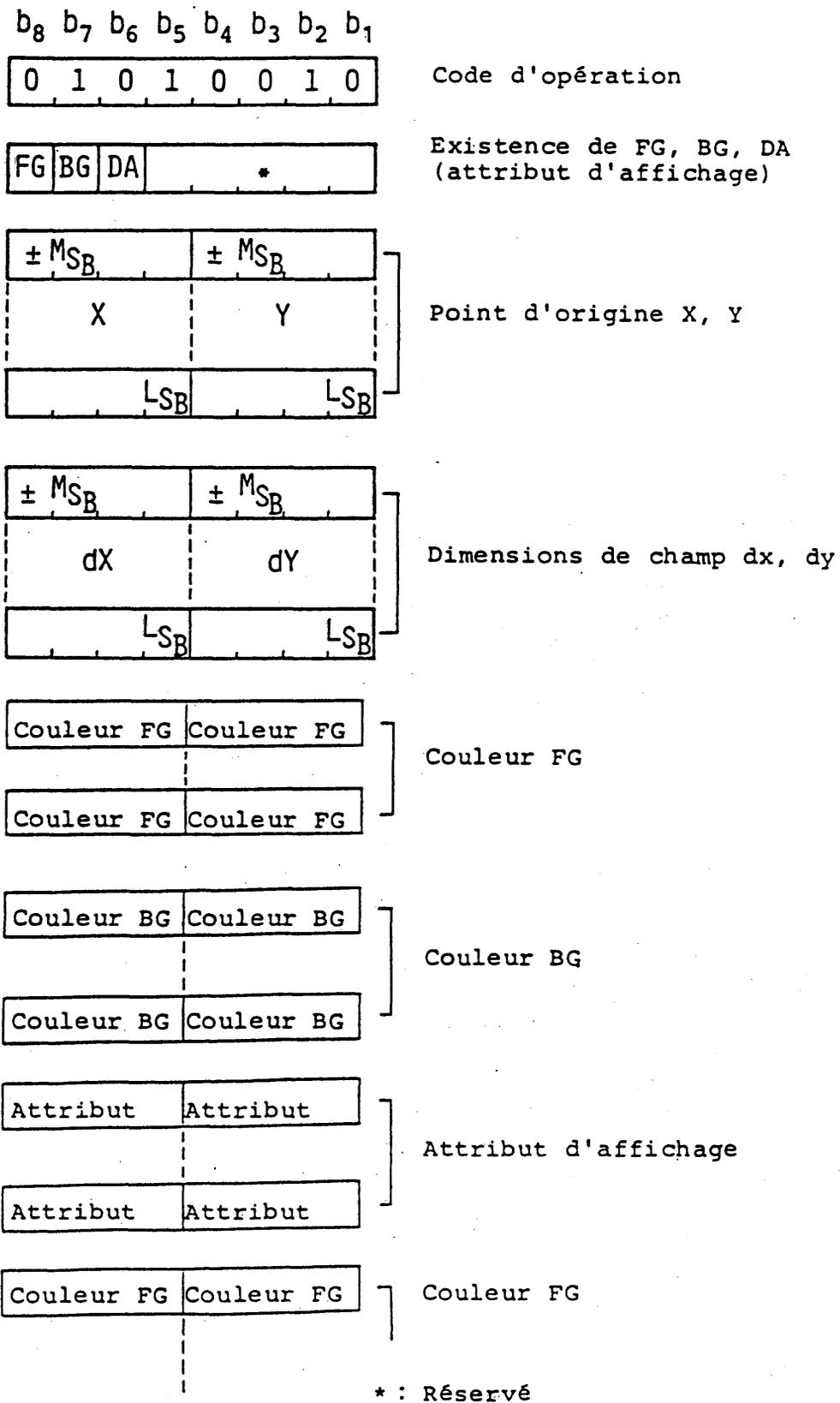


FIGURE 4.11  
BLOC DE COLORATION DE CHAMP

Le premier opérande (format fixe) indique le drapeau d'attribut de la même manière que l'instruction BLOC de COLORATION. Le deuxième et le troisième sont des opérandes à valeur multiple qui spécifient l'emplacement et les dimensions d'un champ rectangulaire dans lequel les données d'attribut doivent être déposées. La position initiale de bloc de coloration est spécifiée en coordonnées absolues par le premier opérande à valeur multiple et la largeur et la hauteur (dx, dy) sont données par le deuxième opérande à valeur multiple. La position initiale de bloc de coloration peut être l'un des quatre sommets du champ car la largeur et la hauteur peuvent être positives ou négatives.

Les données photographiques sont composées de codes de trois-quatre bits successifs qui représentent les données FG, les données BG et les attributs d'affichage. Un quatre-bits pour les données FG ou les données BG est une adresse d'entrée du LUT. Les attributs d'affichage à quatre-bits sont utilisés pour les commandes de clignotement, et leur signification est indiquée dans le tableau 4.4

Lorsque le premier opérande indique que tous les drapeaux sont des «1», les données de couleur FG pour le champ viennent en premier, les données de couleur BG pour le même champ viennent ensuite et les données d'attribut d'affichage pour le même champ suivent.

Lorsque la largeur (dx) d'un champ est positive, la position de dessin se déplace vers la position de pixel suivante à droite après le dépôt des données dans un pixel. Lorsque la largeur (dx) est négative, la position de dessin se déplace vers la gauche.

Lorsque la hauteur (dy) d'un champ est positive, la position de dessin se déplace vers la ligne immédiatement supérieure, après le remplissage d'une ligne complète dans un champ. Lorsque la hauteur (dy) est négative, la position de dessin se déplace vers la ligne immédiatement inférieure. Si les données dépassent le champ défini, les données suivantes sont incorporées dans le champ dont les paramètres sont (X + dx, Y) et (dx, dy).

Si le champ dépasse la limite de plan de coloration de bloc, les données restantes sont rejetées.

Les données d'opérande doivent maintenir une limite d'octet dans le champ d'opérande.

#### 4.8 *BLOC DE COLORATION DE CHAMP AVEC COMPRESSION*

Le format de la PDU est indiqué sur la figure 4.12. Les données photographiques expriment les couleurs et les attributs d'affichage des blocs de coloration dans un champ sur les plans de FG, de BG et d'attribut d'affichage.

Le premier opérande (format fixe) indique le drapeau d'attribut de la même manière que le BLOC de COLORATION.

Le deuxième et le troisième sont des opérandes à valeur multiple qui spécifient l'emplacement d'un champ rectangulaire et les dimensions des blocs de coloration sur lesquels les données d'attribut doivent être déposées. La position initiale de bloc de coloration est spécifiée en coordonnées absolues par le premier opérande à valeur multiple et la largeur et la hauteur (dx, dy) sont données par le deuxième opérande à valeur multiple. La position initiale de bloc de coloration peut être l'un des quatre sommets du champ car la largeur et la hauteur peuvent être positives ou négatives.

Les données photographiques suivant les paramètres se composent de paires de code attribut/passage, le nombre de bits de chaque partie est huit (paire 4 bits-4 bits), de la même manière que la commande de COLORATION AVEC COMPRESSION. Le dernier passage dans le champ défini est codé comme ECL.

La position du bloc de coloration est automatiquement déplacée dans la direction x sur une distance égale à la largeur (dx<sub>0</sub>) du bloc de coloration. Si la dimension dx est positive, la position se déplace vers la droite; si elle est négative, la position se déplace vers la gauche.

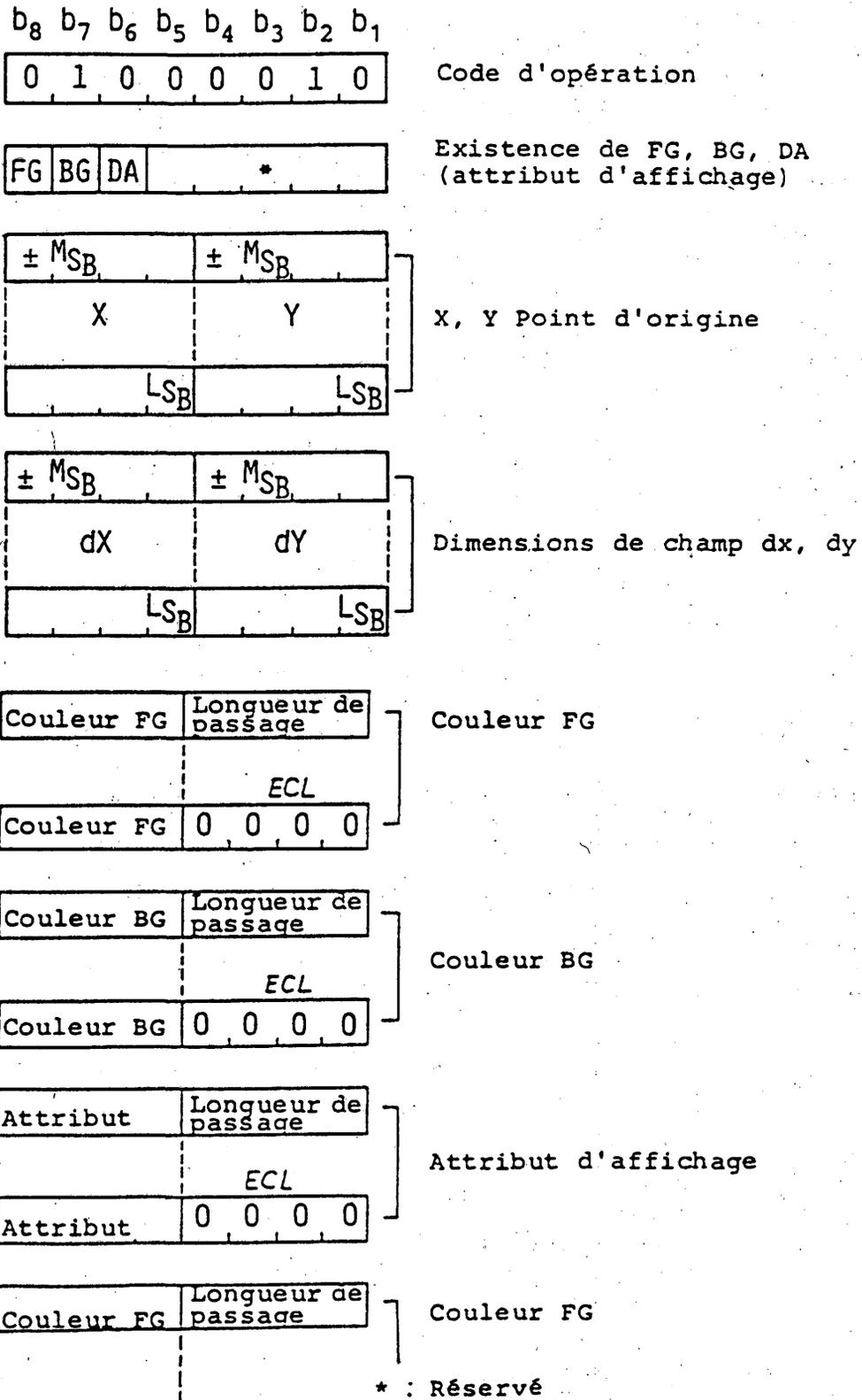


FIGURE 4.12

BLOC DE COLORATION DE CHAMP AVEC COMPRESSION

Si la position du bloc atteint ou dépasse un côté vertical du champ, la coordonnée x du bloc de coloration est déplacée de l'autre côté du champ et la coordonnée y est déplacée sur une distance égale à la hauteur (dy) du bloc de coloration. Si la dimension du champ (dy) est négative, la position du bloc se déplace vers le bas.

Si la position du bloc atteint ou dépasse un côté horizontal du champ, et s'il reste des données photographiques, elles doivent être interprétées comme si elles poursuivent le même code d'opération avec les opérandes de paramètre du champ (x + dx, y), (dx, dy).

#### 4.9 *BLOC DE COLORATION DE FORMAT LIBRE*

Le format PDU est indiqué sur la figure 4.13. Les données photographiques sont exprimées de la même manière que dans la commande de COLORATION DE CHAMP AVEC COMPRESSION, à l'exception de l'affectation des bits d'une partie du code attribut/passage.

Le premier opérande de paramètre de format fixe indique le contenu des données photographiques, la longueur de code FG, BG, DA et l'expression de la longueur de passage. Les bits et leur signification sont indiqués dans le tableau 4.6.

Le code de longueur de passage spécifié par le premier opérande de paramètre est indiqué dans le tableau 4.7.

#### 4.10 *JCDR 1 PHOTO ET JCDR 2 PHOTO*

Les formats PDU sont indiqués sur la figure 4.14. Les données photographiques expriment le modèle des pixels du caractère à définir. L'opérande est un format fixe de trois octets. Le premier opérande de paramètre à format fixe indique le nombre de pixels dans un bloc de caractère pour les directions horizontale comme verticale. Les codes sont les suivants:

7/3: 16 × 24 pixels (taille normale)

7/8: 8 × 24 pixels (taille moyenne)

7/1: 8 × 12 pixels (petite taille).

Le deuxième octet est fixé à 4/1 pour le JCDR 1, ou bien représente le premier code d'octet pour le JCDR 2. Le troisième octet est le code JCDR pour le JCDR 1 et le deuxième code d'octet pour le JCDR 2.

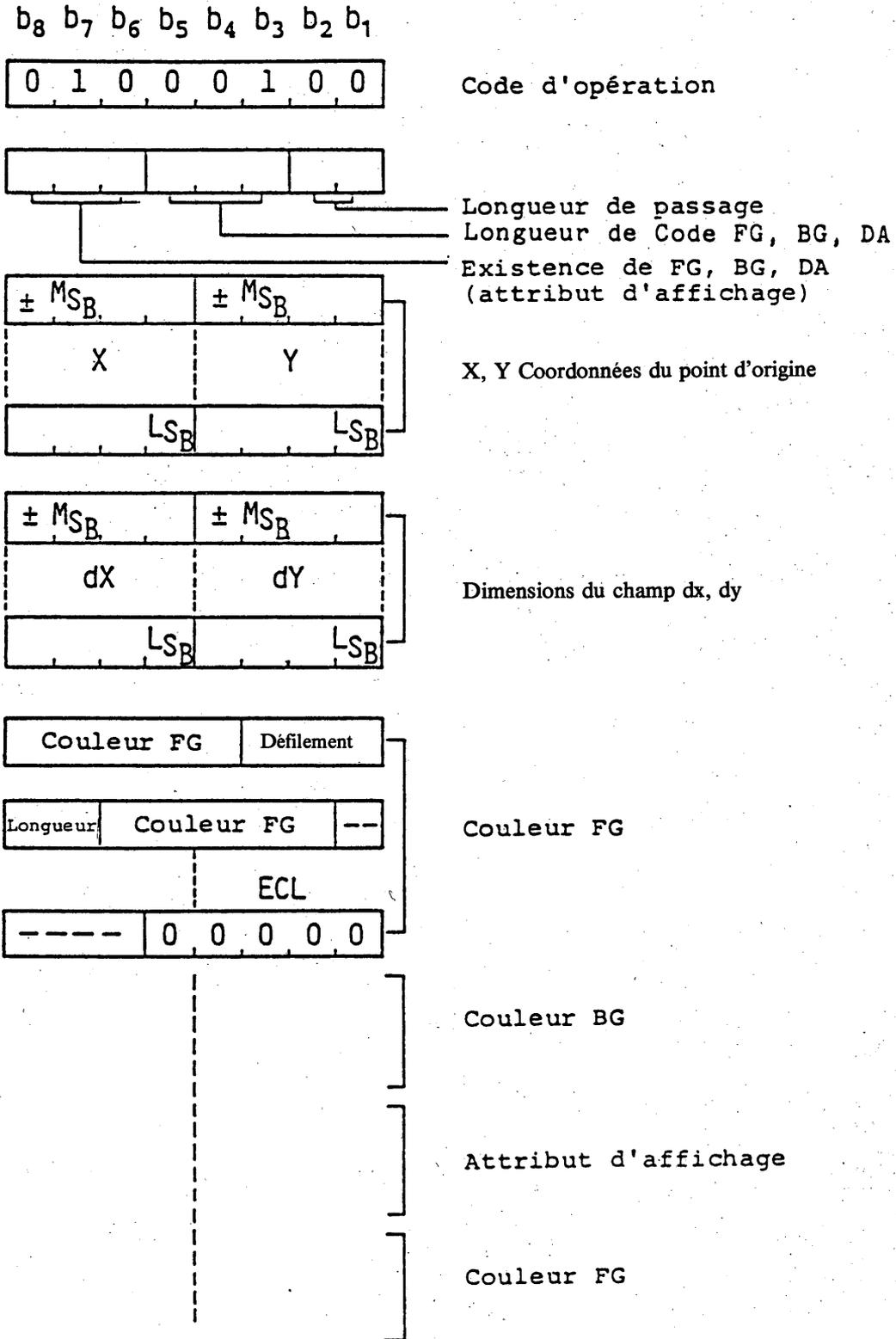


FIGURE 4.13  
 BLOC DE COLORATION DE FORMAT LIBRE

TABLEAU 4.6

**Interprétation d'octet de format fixe de  
BLOC DE COLORATION DE FORMAT LIBRE**

b <sub>8</sub>	b <sub>7</sub>	b <sub>6</sub>	Contenu des données
0	0	1	DA (attributs d'affichage)
0	1	0	BG
0	1	1	BG et DA
1	0	0	FG
1	0	1	FG et DA
1	1	0	FG et BG
1	1	1	FG, BG et DA
b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	Longueur de code FG/BG/DA en 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
b <sub>2</sub>	b <sub>1</sub>	Longueur de code de longueur de passage en bits	
0	0	5	
0	1	6	
1	0	7	
1	1	8	

TABLEAU 4.7

Code de longueur de passage pour BLOC DE COLORATION DE FORMAT LIBRE

Code	Longueur de passage
00-----00	ECL
00-----01	1
00-----10	2
00-----11	3
-	-
-	-
-	-

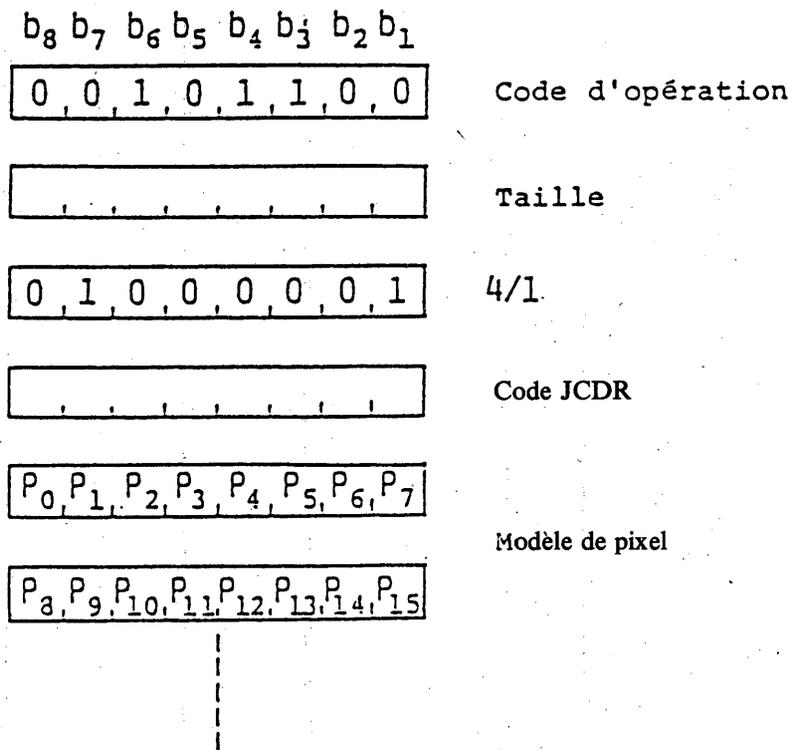


FIGURE 4.14 (1)

JCDR 1 PHOTO

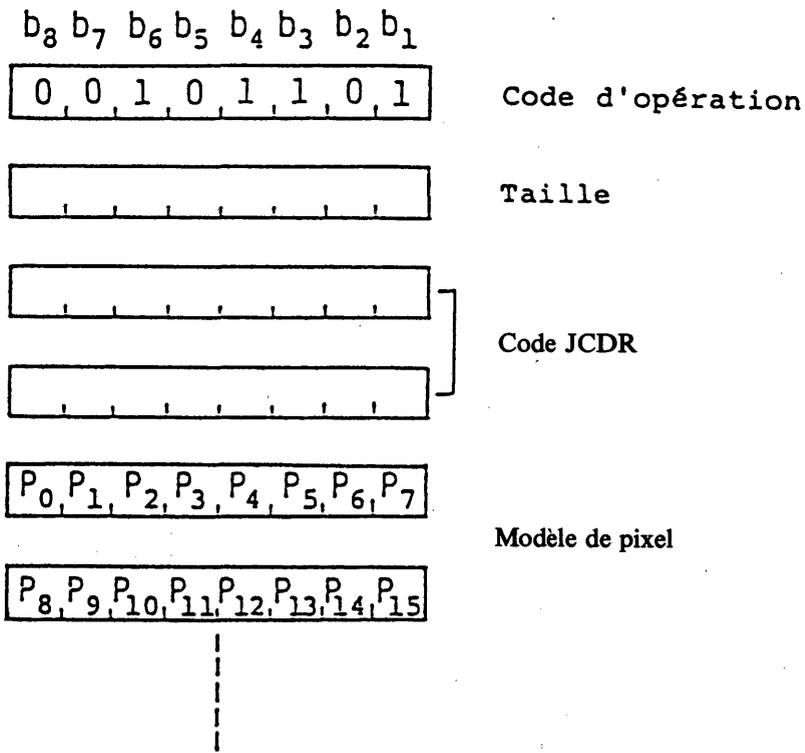


FIGURE 4.14 (2)  
**JCDR 2 PHOTO**

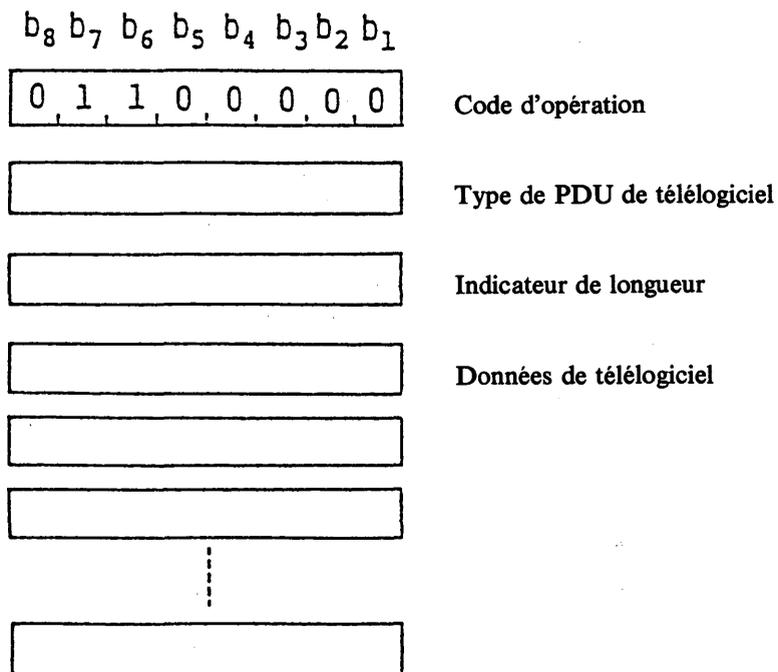


FIGURE 4.15  
**PDU de télégiciel**

#### 4.11 Télégiciel

Le format PDU du télégiciel est indiqué sur la figure 4.15. La partie données correspond aux données de télégiciel proprement dites. Le premier opérande de paramètre à format fixe indique les types de PDU du télégiciel expliqués dans le tableau 4.8.

Normalement, la longueur des données de télégiciel est supérieure à celle des données d'image.

Les erreurs sur les bits de transmission ont des conséquences plus graves sur les données de télégiciel que sur les données d'image.

La transmission des données de télégiciel dans une PDU unique est risquée car les données de télégiciel sont divisées en plusieurs PDU.

Le deuxième opérande de paramètre à format fixe indique la longueur des données de télégiciel séparées.

TABLEAU 4.8

#### Type de PDU de télégiciel

$b_8$	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	
0	0	0	0	0	0	0	0	Première PDU des données de télégiciel
0	0	0	0	0	0	0	1	PDU intermédiaire des données de télégiciel
1	1	1	1	1	1	1	1	Dernière PDU des données de télégiciel
0	0	0	0	0	0	1	0	Réservé
1	1	1	1	1	1	1	0	

## 5 Codage dans le mode musical

### 5.1 Généralités

Les données de note musicale sont transmises dans le mode musical. Bien que les données soient codées en caractère dans le mode musical, la présente norme spécifie le mode séparément du mode code de caractère car le contenu des informations est différent. Un jeu-C et un jeu-G différents de ceux du mode code de caractère sont initialisés sur appel du mode musical.

La figure 5.1 illustre le réglage de table dans le mode musical.

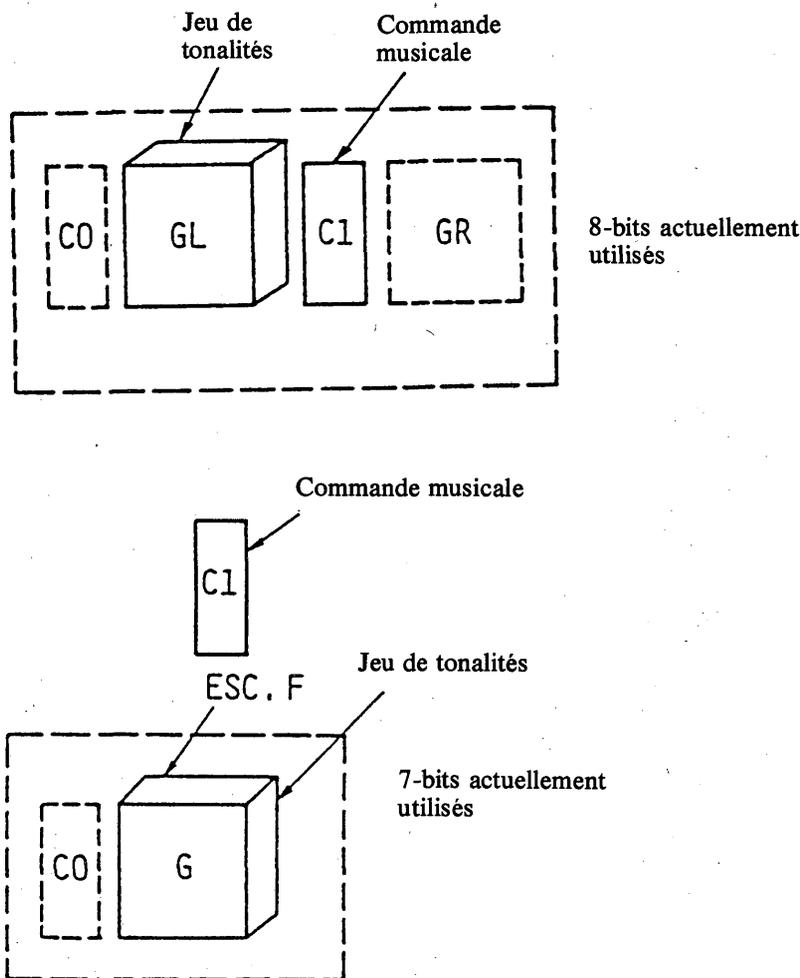


FIGURE 5.1

Réglage de table dans le mode musical

### 5.2 Jeu de tonalités musicales

Le jeu de tonalités musicales est un jeu de code à deux octets qui représente la hauteur et la durée d'une note de musique. Le jeu de tonalités musicales est indiqué dans le tableau 5.1.

TABLEAU 5.1 (1)

Jeu de tonalités musicales (hauteur)

					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	

TABLEAU 5.1 (2)

Jeu de tonalités musicales (durée)

					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	Réservé		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: Unité de temps

Remarque – Les colonnes 2 et 3 sont réservées pour une extension future.

Le premier octet d'un code spécifie la hauteur de la tonalité, définissant quatre-vingt-huit hauteurs de A0 à C8.

La fréquence de hauteur A4 est fixée à 440 Hz: la fréquence de chaque hauteur est exprimée comme:

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

où:

*I* est l'expression décimale du premier octet.

IP (hauteur indéfinie) (lorsque le premier octet est 7/15) est utilisé pour une partie du rythme qui a une hauteur fixe.

RST (lorsque le premier octet est 2/0) est utilisé pour provoquer une pause.

Le deuxième octet détermine la durée d'un son dont la hauteur est spécifiée par le premier octet. La durée du son est exprimée avec un nombre de Tu, Tu étant une longueur unitaire de durée. La valeur de Tu est transmise avant les codes de tonalité par SMC (voir le § 5.3.1.1).

### 5.3 *Jeu de commandes musicales*

Douze codes de commande sont utilisés pour commander la production génération de son. Le jeu de commandes musicales est indiqué dans le tableau 5.2.

#### 5.3.1 *Départ et fin de codes de musique*

##### 5.3.1.1 *SMC (Départ des codes de musique)*

SMC dénote un départ de séquence de code de musique. Cette commande a deux paramètres. Le premier paramètre représente l'identification du système de codage. Le système de codage suivant est le défaut: le numéro d'identification est 3/0. Les autres numéros sont réservés pour une normalisation future.

Le deuxième paramètre Tu représente la longueur unitaire de durée du son en millisecondes. (L'expression de paramètre numérique est décrite au § 5.3.7.)

SMC	<u>3/0</u>	3/11	<u>3/N</u>	3/11
	Identification		Longueur unitaire Tu	

##### 5.3.1.2 *EMC (Fin des codes de musique)*

EMC dénote la fin de la séquence de codes de musique.

#### 5.3.2 *Départ et fin d'une partie*

Chaque information de partie est définie consécutivement, partie par partie. Chaque séquence de définition est séparée par le code de départ de partie (SMP ou SRP) et par le code de fin de partie (EOP).

##### 5.3.2.1 *SMP (Départ de partie de mélodie)*

SMP dénote le départ d'une partie de mélodie et spécifie le timbre joué dans cette partie de mélodie. Le timbre est défini par le paramètre de SMP indiqué dans le tableau 5.3.

SMP	<u>3/P</u>	3/11
	timbre	

TABLEAU 5.2

Jeu de commandes musicales

b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>			
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		

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
	4/8	5/9

TABLEAU 5.3

Timbre dans une partie de mélodie

Timbre (Instrument)	Code	
Violon	3/1	3/0
Guitare (Clavecin)	3/1	3/1
Piano	3/1	3/2
Flûte	3/2	3/0
Clarinette	3/2	3/1
Hautbois	3/2	3/2
Trompette	3/2	3/3
Orgue	3/3	3/0
Vibraphone	3/4	3/0

5.3.2.2 SRP (Départ de partie de rythme)

SRP dénote le départ d'une partie de rythme ou de percussion et spécifie le timbre joué dans cette partie de rythme. Le timbre est défini par le paramètre de SRP indiqué dans le tableau 5.4.

SRP    3/P    3/11  
           |  
       timbre

TABLEAU 5.4

Timbre dans une partie de rythme

Timbre (Instrument)	Code	
Tambour à timbre	3/1	3/0
Grosse caisse	3/1	3/1
Tom-tom	3/2	3/0
Cymbale suspendue (Cymbale haute)	3/3	3/0
Cymbales double à coulisse	3/3	3/1

### 5.3.2.3 EPT (Fin de partie)

EPT dénote la fin d'une partie de mélodie ou d'une partie de rythme.

### 5.3.3 Répétition et saut

Les quatre codes suivants sont utilisés pour exprimer les répétitions ou les sauts. Ils sont suivis d'un ou deux paramètres numériques, M et/ou N.

M indique un nombre de répétitions ou de temps de passage. N représente un numéro d'étiquette tel que le point de départ d'une répétition ou la destination pour un saut.

#### 5.3.3.1 LBL (Etiquette)

Suivi du paramètre N, LBL définit une étiquette comme point de départ d'une destination.

LBL    3/N    3/11  
          |  
          étiquette

#### 5.3.3.2 JMP (Saut)

JMP entraîne le saut vers la partie portant l'étiquette avec le paramètre N.

JMP    3/N    3/11  
          |  
          destination (étiquette)

#### 5.3.3.3 RPT (Répétition)

Suivi de deux paramètres M et N, RPT dénote le passage après M répétitions. Le point de départ de la répétition est indiqué par le deuxième paramètre N.

RPT    3/M    3/11    3/N    3/11  
          |                                   |  
          Nombre de répétitions        Point de départ (étiquette)

#### 5.3.3.4 BRA (Branchement)

Suivi de deux paramètres M et N, BRA dénote un saut vers l'étiquette N, juste après le M<sup>e</sup> passage.

BRA    3/M    3/11    3/N    3/11  
          |                                   |  
          Nombre de passages            Destination (étiquette)

### 5.3.4 SLV (Niveau sonore)

Le niveau sonore (intensité) est dénoté par SLV, suivi d'un paramètre indiqué dans le tableau 5.5. L'intensité du son est spécifiée en huit niveaux, du niveau 1 au niveau 8.

Le niveau 1 indique l'intensité minimale et le niveau 8 l'intensité maximale.

Les différences entre niveaux sonores adjacents sont d'environ 3 dB.

SLV    3/L    3/11  
          |  
          niveau

TABLEAU 5.5

Niveau sonore

Niveau	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

Niveau 1: minimal

Niveau 8: maximal

Niveau 4: défaut

5.3.5 CTM (*Changement de timbre*)

Le changement de timbre dans une partie est dénoté par CTM, suivi d'un paramètre indiqué dans le tableau 5.3.

CTM  $\frac{3/P}{\text{timbre}}$  3/11

5.3.6 LRT (*Longue pause et tonalité*)

LRT est un code qui permet de dénoter une longue tonalité ou une longue pause, entraînant la multiplication par N de la durée de la tonalité définie suivante.

LRT  $\frac{3/N}{\text{multiplicateur}}$  3/11

5.3.7 Paramètre numérique

Les paramètres numériques suivant un code de commande musical sont exprimés en décimal codé binaire. Les caractères 3/0 à 3/9 sont utilisés pour indiquer une valeur et 3/11 est utilisé comme un délimiteur décimal.

Par exemple, la séquence

$\frac{3/1, 3/2, 3/3}{N_1}, \frac{3/11, 3/4, 3/5, 3/6}{N_2}, 3/11$

est interprétée comme

$N_1 = 123$  (décimal)

et  $N_2 = 456$  (décimal)

## Conditions de défaut

Les conditions de défaut des réglages de table, des attributs d'affichage, etc., sont indiquées ci-dessous.

Élément	Condition de défaut
Mode d'instruction	Mode code de caractère
Jeux graphiques	
G0	Jeu Kanji-japonais
G1	Jeu primaire
G2	Jeu d'instructions de commande d'affichage
G3	JCDR 1
Tableau en cours d'usage	
GL	G0
GR	G2
(Dans un environnement à 7 bits)	G0)
Opérande à valeur unique	1 octet
Longueur	
Opérande à valeur multiple	3 octets (mode code de caractère) 2 octets (mode transparent)
Tailles de pel logique	(1/256, - 1/256)
Tailles de bloc de coloration	(4/256, - 4/256)
Tailles de bloc de caractères	(16/256, - 24/256)
Couleur de premier plan	Blanc (7 <sup>e</sup> couleur)
Couleur d'arrière-plan	Transparent (8 <sup>e</sup> couleur)
Contenu du LUT	Spécifié dans le tableau 3.5
Couleur de trame	Bleu d'intensité réduite
Couleur de trame d'en-tête	Bleu d'intensité réduite
Affichage de curseur	Arrêt
Clignotement	Arrêt
Dissimulation	Arrêt
Traçage de ligne	Arrêt
Protection	Arrêt
Défilement	Arrêt
Clignotement	Arrêt
Taille de caractère	Normale
Rotation de texte	0 degré
Trajet de caractère	Vers la droite
Espacement entre caractères	1
Espacement entre rangées	1
Paramètre de mouvement	Mouvement indépendant
Style de curseur	Souligné
Position de curseur	Position de rattachement (en haut à gauche)
Point de dessin	(0,0)
Texture de ligne	Solide
Remplissage de modèle	Solide
Renforcement	Arrêt
Zone active	Zone d'affichage définie
Index de trame	10
Mode d'écriture par superposition	Remplacement
Macros	Null
JCDR 1 et 2	Null

APPENDICE 1

MODÈLE DE RÉFÉRENCE DE SERVICE (SRM)

Le modèle de référence de service (SRM) définit les caractéristiques qui devraient être utilisées dans un terminal ordinaire. Ces ensembles de conditions représentent également les éléments fonctionnels maximaux que le fournisseur de l'information devrait supposer lors du codage de l'information sous forme de texte ou d'image. Le SRM est censé constituer un moyen d'utilisation d'un terminal ou de création de pages. Il n'a cependant rien d'obligatoire.

	Fonction	Condition requise
1.	Séquence d'extension de code	Toutes les séquences d'extension de code définies dans la présente norme devraient être exécutées.
2.	Jeu de commandes C0	Le jeu devrait être exécuté de la manière décrite dans la présente norme.
3.	Jeu de commandes C1	Le jeu devrait être exécuté de la manière décrite dans la présente norme, à l'exception du processus de clignotement qui peut être limité à deux phases. L'intervalle MARCHE et l'intervalle ARRÊT sont chacun d'environ 0,5 seconde.
4.	Répertoire de caractères graphiques	Tous les caractères du jeu primaire, du jeu Kanji-japonais et du jeu Katakana devraient être présentés.
5.	Jeux de mosaïques	Toutes les mosaïques du jeu de mosaïques 1 et du jeu de mosaïques 2 devraient être présentées.
6.	JCDR	Les deux jeux JCDR 1 et JCDR 2 devraient être définissables. Le nombre de caractères définissables de manière simultanée est assujéti à la limitation de la mémoire.
7.	Jeu de macros	Le nombre de macros définissables devrait être de 96, sous réserve de la limitation de la mémoire.
8.	Mémoire nécessaire	Le terminal devrait comporter une mémoire d'au moins 2 Koctets partagée entre les JCDR et les macros.
9.	Jeu d'instructions de commande d'affichage	Les instructions suivantes devraient être utilisées de la manière décrite dans la présente norme.  RÉINITIALISATION P DOMAINE P PEL LOGIQUE MODE D'AFFICHAGE ÉCRAN ÉCRAN D'EN-TÊTE.
10.	Instructions photographiques	Les instructions suivantes devraient être utilisées de la manière décrite dans la présente norme.  CONFIGURATION DE POINTS EN LIGNE CONFIGURATION DE POINTS EN LIGNE AVEC COMPRESSION CONFIGURATION DE POINTS EN CHAMP BLOC DE COLORATION BLOC DE COLORATION AVEC COMPRESSION BLOC DE COLORATION DE CHAMP JCDR 1 PHOTO JCDR 2 PHOTO.
11.	Paramètres d'affichage physique	La résolution devrait être de 248 pixels horizontalement et de 204 pixels verticalement, y compris la «zone d'en-tête».

## APPENDICE 2

### FONCTIONNEMENT DE CS ET DE NSR

CS et NSR rétablissent les attributs suivants:

- Réglage de tableau de jeu graphique
- Domaine
- Tailles des pels logiques
- Tailles des blocs de coloration
- Couleur de premier plan
- Couleur d'arrière-plan
- Affichage du curseur
- Clignotement
- Dissimulation
- Traçage de ligne
- Protection
- Défilement
- Tailles des blocs de caractères
- Attributs d'affichage de texte
- Taille de caractère
- Point de dessin
- Texture de ligne
- Remplissage de modèle
- Renforcement
- Index de trame
- Mode d'écriture par superposition

Les attributs suivants ne sont réglés aux états de défaut que par CS:

- Couleur de trame
- Couleur de trame d'en-tête
- Position du curseur (NSR règle la position du curseur en fonction de ses paramètres)
- Contenu du LUT
- Processus de clignotement
- Zone active
- Contenu de la mémoire de trame
- Définition JCDR
- Définition MACRO

