



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 ROUGE**

---

**FASCICULE VII.3 – ANNEXE III**

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

**RECOMMANDATION T.101, ANNEXE D**

---



**VIII<sup>e</sup> ASSEMBLÉE PLÉNIÈRE**

MALAGA-TORREMOLINOS, 8-19 OCTOBRE 1984

Genève 1985



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

# CCITT

COMITÉ CONSULTATIF  
INTERNATIONAL  
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

**ANNEXE AU LIVRE ROUGE**

---

**FASCICULE VII.3 – ANNEXE III**

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

**RECOMMANDATION T.101, ANNEXE D**

---



**VIII<sup>e</sup> ASSEMBLÉE PLÉNIÈRE**

MALAGA-TORREMOLINOS, 8-19 OCTOBRE 1984

Genève 1985



ANNEXE D

( à la Recommandation T.101 )

SYNTAXE DE DONNEE III

Remarque : La présente syntaxe de données correspond à la syntaxe du protocole de la couche présentation du vidéotex/télétexte (NAPLES) officiellement adoptée par le Canada et les Etats-Unis d'Amérique.

Préface

La présente syntaxe de données détaille la structure de codage à employer dans les services vidéotex/télétexte. Le vidéotex et le télétexte sont respectivement des services bidirectionnels et unidirectionnels par lesquels des usagers ont accès à des "pages" ou "trames" contenant de l'information alphanumérique et des images.

Cette syntaxe de données permet "l'échange aveugle" qui est souhaitable pour le média électronique envisagé pour les services vidéotex/télétexte. Par échange aveugle, on entend que l'expéditeur et le destinataire de l'information n'ont besoin d'établir ni conventions ni modalités de dialogue pour pouvoir échanger utilement de l'information; il leur suffit de respecter la syntaxe de données.

De façon générale, les exigences de conformité prescrites définissent les règles (syntaxe) qui assureront la conformité de l'échange à l'interface de codage, ainsi que l'exécution (sémantique) que doit assurer un processus de présentation conforme à ces exigences. Ces exigences de conformité sont soigneusement détaillées de sorte à permettre l'usage de nombre de techniques de matérialisation différentes. Ainsi, le corps de la syntaxe de données ne précise-t-il pas la résolution du support de visualisation, afin de permettre la présentation d'un même échange au moyen des téléviseurs actuels ou de supports de visualisation à plus haute résolution.

Afin de préciser davantage les ensembles de paramètres de matérialisation spécifiques pour le vidéotex/télétexte, deux modèles de référence de service (MRS) sont donnés à l'annexe D. Un MRS indique la matérialisation minimale d'un appareil récepteur et la matérialisation maximale à laquelle doit s'attendre le fournisseur d'information. Une matérialisation qui satisfait aux exigences d'un MRS est également conforme à la syntaxe de données.

Dans la présente syntaxe de données, les verbes "doit", "devrait" et "peut" expriment respectivement une exigence, une recommandation et une possibilité.

Table des matières	Page
1. Domaine d'application . . . . .	4
2. Définitions . . . . .	5
3. Ouvrages de référence . . . . .	9
4. Structure de codage . . . . .	11
4.1 Modèle de référence (OSI) . . . . .	11
4.2 Vue d'ensemble de la couche présentation . . . . .	11
4.2.1 Généralités . . . . .	11
4.2.2 Système de coordonnées . . . . .	12
4.2.3 Format d'affichage . . . . .	13
4.3 Extension de code . . . . .	14
4.3.1 Généralités . . . . .	14
4.3.2 Extension des codes à 7 éléments . . . . .	15
4.3.3 Extension des codes à 8 éléments . . . . .	20
4.4 ESPACE et OBLITÉRATION . . . . .	21
5. Codage des jeux G . . . . .	23
5.1 Jeu de caractères primaire . . . . .	23
5.2 Jeu de caractères supplémentaire . . . . .	25
5.3 Jeu d'instructions de description de l'image (IDI) . . . . .	27
5.3.1 Généralités . . . . .	27
5.3.2 Fonctions de commande d'attribut . . . . .	35
5.3.2.1 Généralités . . . . .	35
5.3.2.2 DOMAINE . . . . .	35
5.3.2.2.1 Format de la commande . . . . .	35
5.3.2.2.2 Longueur d'opérande à valeur simple . . . . .	36
5.3.2.2.3 Longueur d'opérande à valeurs multiples . . . . .	36
5.3.2.2.4 Mode dimensionnel . . . . .	36
5.3.2.2.5 Longueur d'opérande . . . . .	37
5.3.2.2.6 Pel logique . . . . .	37
5.3.2.3 TEXTE . . . . .	39
5.3.2.3.1 Format de la commande . . . . .	39
5.3.2.3.2 Rotation de caractère . . . . .	40
5.3.2.3.3 Sens du trajet de caractère . . . . .	41
5.3.2.3.4 Espacement des caractères . . . . .	41
5.3.2.3.5 Espacement des rangées . . . . .	42
5.3.2.3.6 Séquence automatique APR APD . . . . .	43
5.3.2.3.7 Attributs de déplacement . . . . .	43
5.3.2.3.8 Styles de curseur . . . . .	44
5.3.2.3.9 Dimensions du champ de caractère . . . . .	45
5.3.2.4 TEXTURE . . . . .	46
5.3.2.4.1 Format de la commande . . . . .	46
5.3.2.4.2 Texture de ligne . . . . .	47
5.3.2.4.3 Rehaussement . . . . .	48
5.3.2.4.4 Motif . . . . .	48
5.3.2.4.5 Taille de masque . . . . .	49

	<u>Page</u>
5.3.2.5 DÉTERMINATION COULEUR . . . . .	50
5.3.2.6 SÉLECTION COULEUR . . . . .	57
5.3.2.7 CLIGNOTEMENT . . . . .	59
5.3.2.8 ATTENTE . . . . .	61
5.3.2.9 INITIALISATION . . . . .	62
5.3.3 Primitives de dessin géométriques . . . . .	65
5.3.3.1 POINT . . . . .	65
5.3.3.2 LIGNE . . . . .	68
5.3.3.3 ARC . . . . .	72
5.3.3.4 RECTANGLE . . . . .	77
5.3.3.5 POLYGONE . . . . .	81
5.3.3.6 INCRÉMENTATION . . . . .	86
5.3.3.6.2 CHAMP . . . . .	86
5.3.3.6.3 INCRÉMENTATION POINT . . . . .	87
5.3.3.6.4 INCRÉMENTATION LIGNE . . . . .	90
5.3.3.6.5 INCRÉMENTATION POLYGONE (rempli) . . . . .	93
5.4 Jeu mosaïque . . . . .	95
5.5 Jeu de macro-instructions . . . . .	97
5.6 Jeu de caractères dynamiquement redéfinissables (JCDR) . . . . .	97
<b>6. Codage des jeux C . . . . .</b>	<b>98</b>
6.1 Jeu de commande C0 . . . . .	98
6.2 Jeu de commande C1 . . . . .	104
<b>7. Répertoire de caractères graphiques . . . . .</b>	<b>112</b>
<b>8. Exigences de conformité . . . . .</b>	<b>133</b>
8.1 Généralités . . . . .	133
8.2 Conformité de l'échange . . . . .	134
8.3 Conformité du processus de présentation . . . . .	134
<b>9. Possibilités de développement . . . . .</b>	<b>138</b>
<b>Annexes</b>	
<b>Annexe A, Modèle d'architecture en couches . . . . .</b>	<b>139</b>
<b>Annexe B, Principe du système de coordonnées . . . . .</b>	<b>142</b>
<b>Annexe C, Extension de code et transformation 7 éléments/8 éléments . . . . .</b>	<b>145</b>
<b>Annexe D, Modèle de référence de service (MRS) général du vidéotex et modèle de référence de service (MRS) général du télétexte . . . . .</b>	<b>147</b>

## 1. Domaine d'application

1.1 Se fondant sur l'architecture définie dans le modèle de référence en couches du CCITT pour l'interconnexion des systèmes ouverts, la présente syntaxe de données décrit les formats, règles et procédures de codage de l'information de texte alphanumérique et d'image dans les applications vidéotex et télétexte. La présente syntaxe de données vise spécialement les protocoles de la couche présentation de l'OSI, ainsi que certaines règles particulières de sémantique concernant la couche application des systèmes vidéotex et télétexte.

1.2 Le système de codage utilisé repose sur la structure établie par l'Avis S.100-1980 (Échange international d'information pour le vidéotex interactif) du Comité consultatif international télégraphique et téléphonique (CCITT). Ce système permet l'emploi de codes à 7 et à 8 éléments. Il fournit aussi un jeu de caractères alphanumériques de texte, un jeu de caractères supplémentaire et un jeu de caractères dynamiquement redéfinissables (JCDR). Des primitives mosaïques et géométriques ainsi qu'un JCDR peuvent servir à l'affichage d'images. Le codage mosaïque est compatible avec l'Avis S.100-1980 du CCITT. Les primitives géométriques constituent des améliorations compatibles des instructions de description de l'image (IDI) applicables à la structure alphagéométrique facultative définie dans l'Avis S.100-1980 du CCITT. Le système se caractérise également par des cartes de couleurs, une largeur de trait variable, des macro-instructions, une mise à l'échelle continue des caractères, des masques de texture programmables, des champs sans protection, un défilement partiel de l'image et un codage par incrémentation permettant d'obtenir des descriptions très compactes de certaines classes d'images.



## **2. Définitions**

**Coordonnées absolues:** paire ou triplet ordonné de nombres affectés d'un signe entre -1 (inclusivement) et 1 (exclusivement) et spécifiant (sous forme de complément à deux) le nouvel emplacement de la position de dessin par rapport à l'origine de l'écran unitaire. Seules des coordonnées absolues non négatives se trouvent à l'intérieur de l'écran unitaire.

**Texte alphanumérique:** langage écrit, comprenant des lettres alphabétiques avec ou sans signes diacritiques, des chiffres et fractions numériques, des signes de ponctuation, des symboles typographiques et des signes mathématiques, ainsi que le symbole ESPACE et des lettres, signes, symboles, etc. spéciaux. Dans la présente norme, les caractères alphanumériques de texte sont désignés par des noms qui visent à rappeler la signification courante des caractères et symboles affichés. Ces noms n'ont pas pour but de spécifier un style particulier, une fonte, une taille de caractère ni une position à l'intérieur d'un champ de caractère.

**Attribut:** paramètre déterminable s'appliquant à l'information alphanumérique de texte ou d'image subséquente.

**Combinaison de bits:** groupe ordonné de bits (chiffres binaires) représentant un caractère ou une fonction de commande.

**Plage périphérique:** zone de l'écran d'affichage physique située à l'extérieur de la plage d'affichage.

**Jeu C (ou jeu de commande):** désigne les deux jeux de commande, C0 et C1, comprenant chacun 32 positions de caractère disposées en deux colonnes de 16 rangées.

**Champ de caractère:** zone rectangulaire à l'intérieur de laquelle un caractère est affiché.

**Extension de code:** technique permettant de transformer le jeu de caractères à adressage absolu d'un code organisé en multiplats afin d'obtenir un jeu élargi à adressage virtuel.

**Tableau de code:** ensemble de règles claires définissant les rapports entre les combinaisons de bits reçues et les caractères de la couche présentation.

**Interface de codage:** interface par laquelle des combinaisons de bits codées sont transférées entre l'équipement récepteur et les supports de communications.

**Adresse de la carte des couleurs:** nombre ordinal associé à chaque pixel d'une image numérique mémorisée et déterminant l'adresse de la carte des couleurs à laquelle se trouve la valeur de la couleur actuelle du pixel. (On parle parfois tout simplement de "couleur" lorsqu'il n'y a aucun risque de confusion.)

**Carte des couleurs:** tableau de recherche utilisé durant la conversion de balayage de l'image numérique pour convertir les adresses de la carte des couleurs en vraies valeurs de couleur.

**Valeur de couleur:** entrée de la carte des couleurs indiquant la vraie couleur du pixel à afficher.

**Symbole composé:** symbole constitué d'une combinaison de deux symboles ou plus dans un seul champ de caractère, par exemple un signe diacritique et une lettre de base.

**Compte tenu de la résolution physique:** signifie que la position de l'information d'affichage est calculée avec suffisamment de précision pour que l'affichage se fasse à un pixel près de la position vraie (voir Annexe B).

**Curseur:** indicateur logique (aux dimensions du champ de caractère) de la position à laquelle le caractère suivant doit être appliqué sur l'écran. Cette position peut être marquée d'un symbole ou non.

**Désigner:** identifier un jeu déterminé du répertoire des jeux G comme jeu G0, G1, G2 ou G3.

**Plage d'affichage:** partie rectangulaire de l'écran d'affichage physique dans laquelle l'information codée conformément à la présente norme est rendue visible. La plage d'affichage ne comprend pas la plage périphérique.

**Position de dessin:** indicateur logique de la position à laquelle doit commencer l'exécution de la primitive graphique géométrique suivante. Cette position n'est normalement pas marquée d'un symbole.

**Jeu de caractères dynamiquement redéfinissables (JCDR):** jeu G renfermant des caractères définissables dont les profils peuvent être téléchargés à partir de l'ordinateur central.

**Séquence d'échappement:** chaîne de deux combinaisons de bits ou plus commençant par le caractère ÉCHAPPEMENT (ESC). Une séquence d'échappement à trois caractères, comportant un caractère intermédiaire (I) et un caractère final (F), sert principalement à désigner un jeu de code à 94 ou 96 caractères comme l'un des quatre jeux G actifs. Une séquence d'échappement à deux caractères ne comporte que le caractère final (F) et constitue une des méthodes permettant d'appeler les jeux de code dans le tableau effectif. Les formats et règles concernant l'utilisation des séquences d'échappement sont décrits dans la norme ISO 2022-1982.

**Caractère final:** dernier caractère d'une séquence d'échappement.

**Jeu G:** un des quatre jeux G, soit G0, G1, G2 et G3, comprenant chacun 94 ou 96 positions de caractère disposées en six colonnes de seize rangées.

**Répertoire des jeux G:** ensemble des jeux de code disponibles pouvant être désignés pour devenir l'un des jeux G.

**Primitive graphique géométrique:** algorithme de dessin d'image mémorisé localement et pouvant être appelé par un code d'opération et ses opérandes.

**Répertoire des caractères graphiques:** liste des caractères graphiques définis dans la présente syntaxe de données, y compris les lettres accentuées et les caractères obtenus par la composition de deux symboles graphiques ou plus.

**Lié à la matérialisation:** s'applique à une caractéristique qui peut être spécifiée de façon plus complète dans un modèle de référence de service ou par un concepteur, dans les limites imposées par la présente syntaxe de données.

**Effectif:** s'applique aux jeux de code ou aux attributs utilisés pour interpréter ou caractériser les commandes reçues subséquentement.

**Caractère intermédiaire:** tout caractère situé entre le caractère d'échappement et le caractère final dans une séquence d'échappement.

**Appeler:** faire en sorte qu'un jeu de code désigné soit représenté par les combinaisons de bits du tableau effectif prescrit.

**Couche:** chacun des modules du modèle de référence pour l'interconnexion des systèmes ouverts (OSI).

**Inversion avec verrouillage:** appel dans le tableau effectif d'un jeu de code qui reste effectif jusqu'à ce qu'un autre jeu de code soit appelé à sa place.

**Élément d'image logique (pel logique):** structure géométrique associée à une position de dessin et dont la taille détermine la largeur du trait des primitives graphiques. Bien que les termes "pixel" et "pel" soient ordinairement synonymes, ils sont respectivement utilisés dans la présente syntaxe de données pour désigner les éléments d'image physiques et les éléments d'image logiques.

**Macro-code:** chaîne composée de codes de présentation mémorisée localement et représentée par un seul code de caractère appelé macro-instruction. Lorsque la macro-instruction est utilisée, c'est la chaîne mémorisée localement qui est traitée.

**Mosaïque:** matrice rectangulaire d'éléments prédéterminés qui peuvent servir à créer des images par blocs.

**Noir nominal:** couleur noire (zéros seulement) en mode de couleur zéro, ou couleur se trouvant à l'adresse zéro de la carte des couleurs dans les modes de couleur 1 et 2.

**Blanc nominal:** couleur blanche (uns seulement) en mode de couleur zéro, ou couleur se trouvant à l'adresse 011...1 de la carte des couleurs dans les modes de couleur 1 et 2.

**Sans espacement:** s'applique à un caractère dont la réception ne fait pas automatiquement avancer le curseur après l'affichage du caractère.

**Code d'opération:** caractère comportant un multiplét et permettant d'amorcer l'exécution d'une primitive géométrique ou d'une opération de commande mémorisée localement. Un code d'opération peut être suivi d'aucun, d'un ou de plusieurs opérandes.

**Opérande:** chaîne d'un ou de plusieurs multiplats provenant du champ des données numériques du jeu de code IDI et servant à spécifier les paramètres de commande, d'attribut ou de coordonnée associés à un code d'opération.

**Élément d'image physique (pixel):** la plus petite unité affichable sur un appareil donné.

**Information d'image:** information d'affichage résultant de l'application des primitives géométriques, des éléments mosaïques et du JCQR.

**Instruction de description de l'image (IDI):** commande composée d'un code d'opération suivi d'aucun, d'un ou de plusieurs opérandes, qui constituent une instruction exécutable de dessin ou de commande de l'image.

**Couche présentation:** sixième des sept couches de protocole définies par le modèle de référence pour l'interconnexion des systèmes ouverts. L'utilisation de la couche présentation dans la présente syntaxe de données vise principalement le codage de l'information de texte, de graphique et de commande d'affichage.

**Protocole:** ensemble de formats, de règles et de procédures gérant l'échange d'information entre processus d'une même couche.

**Appareil récepteur:** matériel pouvant recevoir des combinaisons de bits codées, par exemple au moyen de télécommunications ou d'échanges physiques de supports de mémorisation.

**Coordonnées relatives:** paire ou triplet ordonné de nombres affectés d'un signe entre -1 (inclusivement) et 1 (exclusivement) et spécifiant (sous forme de complément à deux) le nouvel emplacement de la position de dessin par rapport à l'emplacement précédent dans le cas d'une primitive géométrique, ou encore les dimensions d'un champ donné dans le cas d'une instruction de commande.

**Modèle de référence de service (MRS):** spécifications de l'ensemble minimal de caractéristiques que doit matérialiser un appareil récepteur pour satisfaire aux exigences d'un service particulier, ainsi que de l'ensemble maximal de caractéristiques que devrait offrir un fournisseur d'information pour le codage de l'information de texte et d'image.

**Inversion unique:** appel d'un jeu de code dans le tableau effectif influant seulement sur l'interprétation de la combinaison de bits reçue subséquentement. L'interprétation revient ensuite automatiquement au contenu antérieur du tableau. (L'inversion unique est également appelée "inversion sans verrouillage".)

**Avec espacement:** s'applique à un caractère dont la réception fait automatiquement avancer le curseur après l'affichage du caractère.

**Écran unitaire:** espace adressable de l'affichage logique, dans lequel toutes les opérations de dessin sont exécutées et les caractères alphanumériques sont introduits. Les dimensions de l'écran unitaire vont de 0 (inclusivement) à 1 (exclusivement) sur les axes horizontal (X), vertical (Y) et de profondeur (Z). (Le dernier axe n'est utilisé qu'en mode tridimensionnel.)

### **3. Ouvrages de référence**

**3.1** Lorsque la présente norme renvoie aux publications suivantes, elle renvoie à l'édition indiquée ci-dessous et à tous les modificatifs s'y rattachant.

#### **3.2 Normes ANSI\***

ANSI X3.4-1977

American National Standard Code for Information Interchange (ASCII).

ANSI X3.41-1974

American National Standard Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange.

ANSI X3.110-1983/CSA T500-1983

Videotex/Teletext Presentation Level Protocol Syntax (North American PLPS)

#### **3.3 Normes ACNOR\*\***

Z243.4-1973

Jeux de caractères codés à 7 éléments pour l'échange d'information entre matériels de traitement de l'information.

Z243.35-1976

Code Extension Techniques for Use with the 7-Bit Coded Character Sets of CSA Standard Z243.4-1973.

CSA T500-1983/ANSI X3.110-1983

Syntaxe du protocole de la couche présentation du vidéotex/télétexte (SPCP nord-américaine)

#### **3.4 Rapport du CCIR+**

957-octobre 1981

Caractéristiques des systèmes télétexte, document 11/5001-E.

#### **3.5 Avis du CCITT++**

F.300-1980

Service vidéotex.

S100-1980

Échange international d'information pour le vidéotex interactif.

V.3-1972

Alphabet international No.5

#### **3.6 Ministère des Communications, Canada**

Service de réglementation des télécommunications,

Cahier des charges sur la radiodiffusion CR-14, juin 1981.

#### **3.7 Avis de l'EIA\*\*\*/CCVCs**

1983

North American Basic Teletext Specification (NABTS).

### 3.8 Normes ISO+++

646-1982

Traitement de l'information - Jeu ISO de caractères codés à 7 éléments pour l'échange d'information.

2022-1982

Traitement de l'information - Jeux ISO de caractères codés à 7 et à 8 éléments - Techniques d'extension de code.

2375-1980

Traitement de l'information - Procédure pour l'enregistrement des séquences d'échappement.

DIS 6937/1-1982

Traitement de l'information - Jeux de caractères codés pour la transmission de textes - Partie 1: Introduction générale.

DIS 6937/2-1982

Traitement de l'information - Jeux de caractères codés pour la transmission de textes - Partie 2: Caractères alphabétiques latins et graphiques non alphabétiques.

DIS 7498-1983

Traitement de l'information - Modèle de référence de base pour l'interconnexion des systèmes ouverts.

\*American National Standards Institute.

\*\*Association canadienne de normalisation.

+Comité consultatif international des radiocommunications.

++Comité consultatif international télégraphique et téléphonique.

\*\*\*Electronic Industries Association.

+++Organisation internationale de normalisation.

sComité consultatif sur le vidéotex canadien.

**Note :** Les lettres "DIS" désignent un projet de Norme internationale, qui faire l'objet de révision.

#### 4. Structure de codage

**4.1 Modèle de référence (OSI).** La structure de codage décrite dans la présente norme porte essentiellement sur la couche présentation du modèle de référence à sept couches de l'ISO pour l'interconnexion des systèmes ouverts. Ce modèle de référence est décrit dans l'Avis X.200 du CCITT et dans la norme ISO DIS 7498-1983, Traitement de l'information - Modèle de référence de base pour l'interconnexion des systèmes ouverts (voir Annexe A).

#### 4.2 Vue d'ensemble de la couche présentation

**4.2.1 Généralités.** La présente norme se fonde sur les principes d'extension de code exposés dans la syntaxe de données ISO 2022-1982\*, ainsi que sur les normes et avis existant actuellement à l'échelle nationale et à l'échelle internationale. Les relations entre les diverses normes de codage et la façon dont celles-ci s'intègrent pour créer une syntaxe de données unifiée sont décrites ci-dessous.

Dans la méthode de description des caractères alphanumériques et de l'information d'image par codage des caractères, des codes de caractère particuliers sont identifiés par une séquence codée de 8 bits ou éléments, dont 7 servent d'index à un tableau de code de 128 caractères et le huitième permet l'extension à un autre tableau de code de 128 caractères, comme le décrit plus loin la présente syntaxe de données, ou l'exécution de fonctions dans d'autres couches de protocole, par exemple le contrôle de parité.

Le tableau de code des caractères se présente normalement sous forme de 8 colonnes de 16 rangées; les bits b7, b6 et b5 désignent les colonnes tandis que les bits b4, b3, b2 et b1 désignent les rangées du tableau (voir figure 3). Ce format général est utilisé tout le long de la présente syntaxe de données. Sur les figures, les bits sont numérotés de b1 à b8, le bit b1 occupant la position de poids faible. Voir la figure 1. Le tableau de code peut se subdiviser en secteurs, comme le décrit l'alinéa 4.3.

\*L'ISO prépare actuellement un projet de supplément à la norme 2022-1982, ainsi qu'une révision de la dite norme, qui porte notamment sur la définition et l'extension de code des jeux G de 96 caractères.

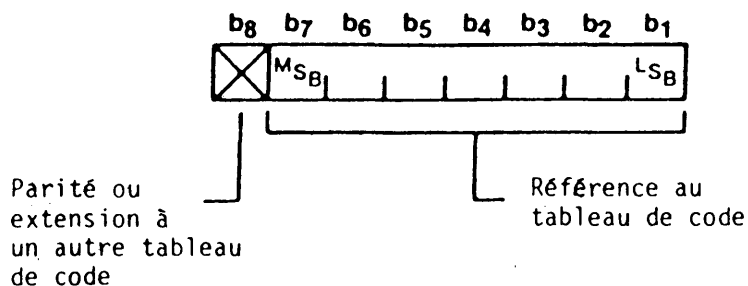


Figure 1

Format de codage

Le codage des caractères alphanumériques repose sur les Avis V.3-1972 et S.100-1980 du CCITT ainsi que sur la norme 646-1982 de l'ISO (voir 5.2).

Le codage de l'information d'image repose sur:

- (1) des instructions de description de l'image (IDI) (voir 5.3), qui se fondent sur des améliorations apportées au système alphagéométrique facultatif décrit dans les Avis S.100-1980 et F.300-1980 du CCITT;
- (2) un jeu mosaïque (voir 5.4), qui se fonde sur la réunion des deux tableaux mosaïques décrits dans les Avis S.100-1980 et F.300-1980 du CCITT;
- (3) un jeu de macro-instructions (voir 5.5); et
- (4) un jeu de caractères dynamiquement redéfinissables (JCDR) (voir 5.6).

**4.2.2 Système de coordonnées.** Le système de coordonnées utilisé dans la présente syntaxe de données repose sur le concept abstrait d'un espace cartésien tridimensionnel à coordonnées unitaires. Ce système n'est donc aucunement lié aux contraintes imposées par le matériel d'affichage. Les coordonnées sont celles de largeur (X), de hauteur (Y) et de profondeur (Z), et la plage des valeurs que peut prendre chaque coordonnée s'étend de 0 (inclusivement) à 1 (exclusivement). À noter que la coordonnée Z est significative uniquement dans le cas des appareils récepteurs pouvant fonctionner en mode tridimensionnel. Son intégration aux structures de données n'a pour but que de compléter le système logique et de faciliter l'introduction harmonieuse de cette nouvelle fonction lorsque la technologie sera disponible. La présente syntaxe de données présuppose l'emploi du mode bidimensionnel, la normalisation complète du mode tridimensionnel étant remise à plus tard. En général, les descriptions viseront donc un plan bidimensionnel (X, Y) de l'espace à  $Z = 0$ . Ce plan sera appelé écran unitaire. La valeur  $Z = 0$  correspond à l'éloignement maximal par rapport à l'utilisateur.

Le dessin des caractères alphanumériques et des images se fait toujours à l'intérieur de l'écran unitaire. L'écran unitaire est visible dans la plage d'affichage, zone rectangulaire de l'écran d'affichage physique de l'appareil. Le coin inférieur gauche de l'écran unitaire est l'origine (0, 0), et coïncide avec le coin inférieur gauche de la plage d'affichage. Tandis que toute la plage d'affichage reste constamment visible, la partie de l'écran unitaire visible dans la plage d'affichage est liée à la matérialisation. À noter qu'un dessin peut toujours être effectué n'importe où sur l'écran unitaire, mais que seule la partie de l'écran unitaire coïncidant avec la plage d'affichage est visible.

À titre d'exemple, les tubes à rayons cathodiques des téléviseurs comportent habituellement un écran d'affichage physique dont le rapport de forme est d'environ 4/3 (largeur/hauteur). Si la plage d'affichage d'un tel appareil est caractérisée par le même rapport de forme de 4/3, la partie visible de l'écran unitaire s'étend alors de 0 (inclusivement) à 1 (exclusivement) sur l'axe des X, et de 0 (inclusivement) à environ 0.75 sur l'axe des Y (voir figure 2).



La plage périphérique ne fait pas partie de la plage d'affichage, et aucune partie de l'écran unitaire n'est visible dans la plage périphérique. À noter que l'écran d'affichage physique peut également servir à d'autres processus d'affichage liés à la matérialisation. Ces processus ne relèvent pas du domaine d'application de la présente syntaxe de données.

**Note:** Le fournisseur peut toujours présupposer que toute la plage d'affichage est disponible. C'est l'appareil récepteur qui doit assurer à l'utilisateur la visibilité de toute la plage d'affichage.

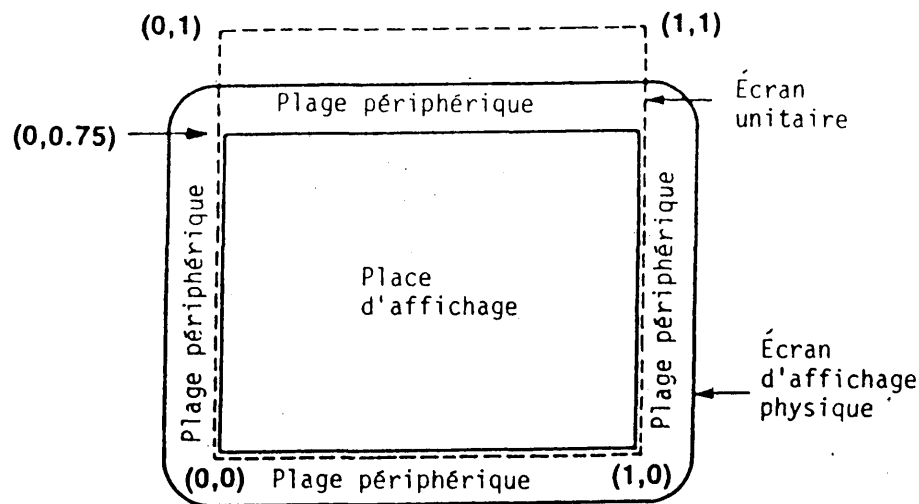


Figure 2

Principe de l'écran unitaire

**4.2.3 Format d'affichage.** Il n'existe aucun lien de dépendance spécial quant à l'ordre dans lequel les primitives de dessin se présentent. Les images sont constituées à partir d'une séquence de commandes de dessin, les primitives étant superposées à celles des commandes précédentes. De cette façon, des images sont constituées par couches. Lorsqu'une commande de dessin ou un caractère alphabétique subséquent vise un pixel donné de l'affichage, toute commande antérieure visant ce pixel se trouve annulée. Plus précisément, cela signifie que des affichages complexes d'images ou de caractères alphabétiques peuvent être composés par la superposition de caractères multiples et de primitives de dessin. La précision du cadrage des caractères alphanumériques et des images ainsi que de la superposition de ces modes de dessin doit être maintenue à un pixel près, compte tenu de la résolution physique.

## 4.3 Extension de code

### 4.3.1 Généralités

4.3.1.1 La méthode d'extension de code utilisée dans la présente syntaxe de données se fonde sur les techniques d'extension de code spécifiées dans la norme ISO 2022-1982.

Cette méthode permet d'effectuer des "désignations" dans le répertoire des jeux et des "appels" dans le tableau effectif, un multiplet spécifié de données codées servant de pointeur dans un tableau de code combiné constitué des jeux C et G. Dans la plupart des applications, le tableau effectif ne comprend pas suffisamment de caractères disponibles, de sorte que la possibilité de permutation des jeux G et C a dû être intégrée dans la structure.

4.3.1.2 Conformément à la norme ISO 2022-1982, tout le système de codage décrit dans la présente syntaxe de données peut faire l'objet de désignations et d'appels par "code complet" au moyen de la séquence d'échappement ESC 2/5 F<sub>1</sub>, où F<sub>1</sub> représente le caractère final qui doit être assigné par l'autorité d'enregistrement de l'ISO ou du CCITT, en conformité avec la norme ISO 2375-1980, Traitement de l'information - Procédure pour l'enregistrement des séquences d'échappement. Pour être conformes à la présente norme, les systèmes n'ont pas nécessairement à utiliser cette séquence d'échappement, sauf en cas d'échange d'information avec d'autres services. Dans ce code complet, une attention spéciale doit être portée aux jeux suivants:

G0: Jeu G à 94 positions de code  
G1: Jeu G à 94 ou 96 positions de code  
G2: Jeu G à 94 ou 96 positions de code  
G3: Jeu G à 94 ou 96 positions de code

Un jeu G à 94 positions de code est un jeu qui ne comprend pas les positions de code 2/0 et 7/15. Lorsqu'un tel jeu est appelé aux colonnes 2 à 7, ces deux positions doivent respectivement signifier ESPACE et OBLITÉRATION.

Par ailleurs, un jeu G à 96 positions de code est un jeu dans lequel les positions 2/0 et 7/15 ont des significations autres que ESPACE et OBLITÉRATION.

La désignation et l'appel de ce "code complet" prennent fin par une séquence différente ESC 1 F<sub>2</sub> (assignation par l'autorité d'enregistrement du CCITT ou de l'ISO ou normalisation par l'ISO) ou par la désignation et l'appel d'un autre "code complet".

4.3.1.3 Quatre jeux G et deux jeux C sont désignés en tout temps, c'est-à-dire que l'un ou l'autre des quatre tableaux (G0, G1, G2 et G3) peut devenir le tableau effectif par suite d'une simple séquence d'appel. Les séquences d'appel peuvent s'exécuter avec ou sans verrouillage. Les jeux G0, G1, G2 et G3 servent de fenêtres dans lesquelles peuvent être désignés des jeux de code provenant du répertoire des significations des jeux G. Dans l'état par défaut, G0 renferme le jeu de caractères primaire, G1 renferme le jeu des IDI, G2 renferme le jeu de caractères supplémentaire et G3 renferme le jeu mosaïque. Une séquence de désignation permet de donner une nouvelle signification à une fenêtre de jeu de code.

**4.3.1.4** Le choix d'un code à 7 ou à 8 éléments peut être établi explicitement, changé pour un service particulier, ou établi implicitement par entente préalable.

Le tableau effectif est structuré en jeux C de 32 positions de code et en jeux G de 94 ou 96 positions de code. Le contenu de ces jeux s'applique aux codes à 7 ou à 8 éléments. Les jeux sont manipulés afin de fournir un adressage virtuel comportant plus que les 128 ou 256 positions respectivement disponibles dans les codes à 7 et à 8 éléments.

**4.3.2 Extension des codes à 7 éléments.** La figure 3 illustre un tableau effectif à 128 positions de code. Chaque combinaison de bits d'entrée est décodée en fonction du contenu actuel de ce tableau, ou est utilisée pour modifier le contenu du tableau. Le tableau comme tel est structuré en 8 colonnes de 16 rangées, les bits 1 à 4 indiquant le numéro des rangées et les bits 5 à 7 le numéro des colonnes. Le tableau effectif renferme le jeu C0 aux colonnes 0 et 1. Cinq caractères de ce jeu, ÉCHAPPEMENT (ESC ou 1/11, c'est-à-dire colonne 1, rangée 1), EN CODE (SI ou 0/15), HORS CODE (SO ou 0/14), INVERSION UNIQUE DEUX (SS2 ou 1/9), et INVERSION UNIQUE TROIS (SS3 ou 1/13), sont utilisés pour établir le contenu des six autres colonnes du tableau effectif. La façon dont s'effectue ce processus est illustrée graphiquement sur la figure 4 et décrite ci-dessous.

Un seul autre jeu de commande actif, soit le jeu C1, et quatre jeux G actifs, soit les jeux G0, G1, G2 et G3, sont également définis. Le contenu du jeu C1 est décrit en 6.2. Le contenu des jeux G0, G1, G2 et G3 peut être sélectionné de façon dynamique dans le répertoire des jeux G par l'emploi de séquences d'échappement. Ces séquences prennent la forme ESC I F, où I est le caractère intermédiaire et F le caractère final. Le caractère intermédiaire détermine quel jeu doit être changé (nouvelle désignation).

La norme ISO 2022-1982 stipule que la syntaxe d'une séquence d'échappement est ESC I...IF, où I...I représente les cas d'aucune, d'une ou de plusieurs occurrences de caractères intermédiaires dans la plage 2/0 à 2/15, et F désigne un seul caractère final dans la plage 3/0 à 7/14. La présence de toute autre combinaison de bits dans une séquence d'échappement doit entraîner la fin de la séquence d'échappement partielle, qui doit rester sans effet, et doit entraîner l'exécution de la combinaison de bits en question.

Le caractère final détermine quel jeu du répertoire doit être sélectionné. Le tableau 1 montre les paires de caractères I et F assignées à chaque jeu C et G. À titre d'exemple, le caractère F du jeu de caractères primaire est 4/2, et le caractère I du jeu G0 est 2/8. La séquence d'échappement à trois caractères, ESC 2/8 4/2, désigne donc le jeu de caractères primaire comme jeu G0 actuel.

Tableau 1

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

Séquence d'échappement	Jeu à désigner
Jeux de commande:	
ESC 2/1 F <sub>3</sub>	Jeu C0
ESC 2/2 F <sub>4</sub>	Jeu C1
Jeux de 94 caractères:	
ESC I 4/2	Jeu de caractères primaire
ESC I 7/12	Jeu de caractères
où I représente respectivement 2/8, 2/9, 2/10 et 2/11 pour G0, G1, G2 et G3	supplémentaire
Jeux de 96 caractères:	
ESC I 5/7	Jeu ID1
ESC I 7/13	Jeu mosaïque
ESC I 7/10	Jeu de macro-instructions
ESC I 7/11	JCDR
où I représente respectivement 2/9, 2/10 et 2/11 pour G1, G2 et G3*	
et I représente aussi respectivement 2/13, 2/14 et 2/15 pour G1, G2 et G3.	

\*Deux caractères I entraînent le changement de désignation de chacun des trois jeux G, G1, G2 et G3. Il est possible que ce double codage soit supprimé de la présente syntaxe de données à l'occasion d'une révision éventuelle.

Les séquences de désignation et d'appel des jeux C0 et C1 sont respectivement ESC 2/1 F<sub>3</sub> et ESC 2/2 F<sub>4</sub>, où F<sub>3</sub> et F<sub>4</sub> doivent être assignés par le CCITT ou l'autorité d'enregistrement de l'ISO, en conformité avec la norme ISO 2375-1980. Toutes les séquences de désignation des jeux G et C1 définies dans la présente syntaxe de données doivent être matérialisées. D'autres jeux G et C1 définis selon les règles de la norme ISO 2022-1982 peuvent également être matérialisés. Toutes les autres séquences de désignation doivent désigner un jeu G ou C1 vide. Le changement de désignation du jeu C0 n'est pas permis dans le cadre de la présente syntaxe de données, et les séquences d'échappement visant un tel changement doivent rester sans effet. Un jeu vide est un jeu dans lequel toutes les positions de code sont traitées par des opérations vides.

Le caractère EN CODE (SI) permet d'appeler le jeu G0 actuel dans le tableau effectif, où il demeure jusqu'à l'opération de commande suivante (il s'agit donc d'un appel avec verrouillage). Le caractère HORS CODE (SO) permet d'appeler et de verrouiller le jeu G1 actuel dans le tableau effectif. La séquence INVERSION AVEC VERROUILLAGE DEUX (LS2) est utilisée pour appeler et verrouiller le jeu G2 dans le tableau effectif. La séquence INVERSION AVEC VERROUILLAGE TROIS (LS3) est utilisée pour appeler et verrouiller le jeu G3 dans le tableau effectif. Le tableau 2 montre le codage des fonctions d'inversion.

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								
<table> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> </table>									0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7									

b4	b3	b2	b1	Rangée
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

SP

Jeu CO

Jeu G

DEL

**Figure 3**  
Tableau effectif de code à 7 éléments

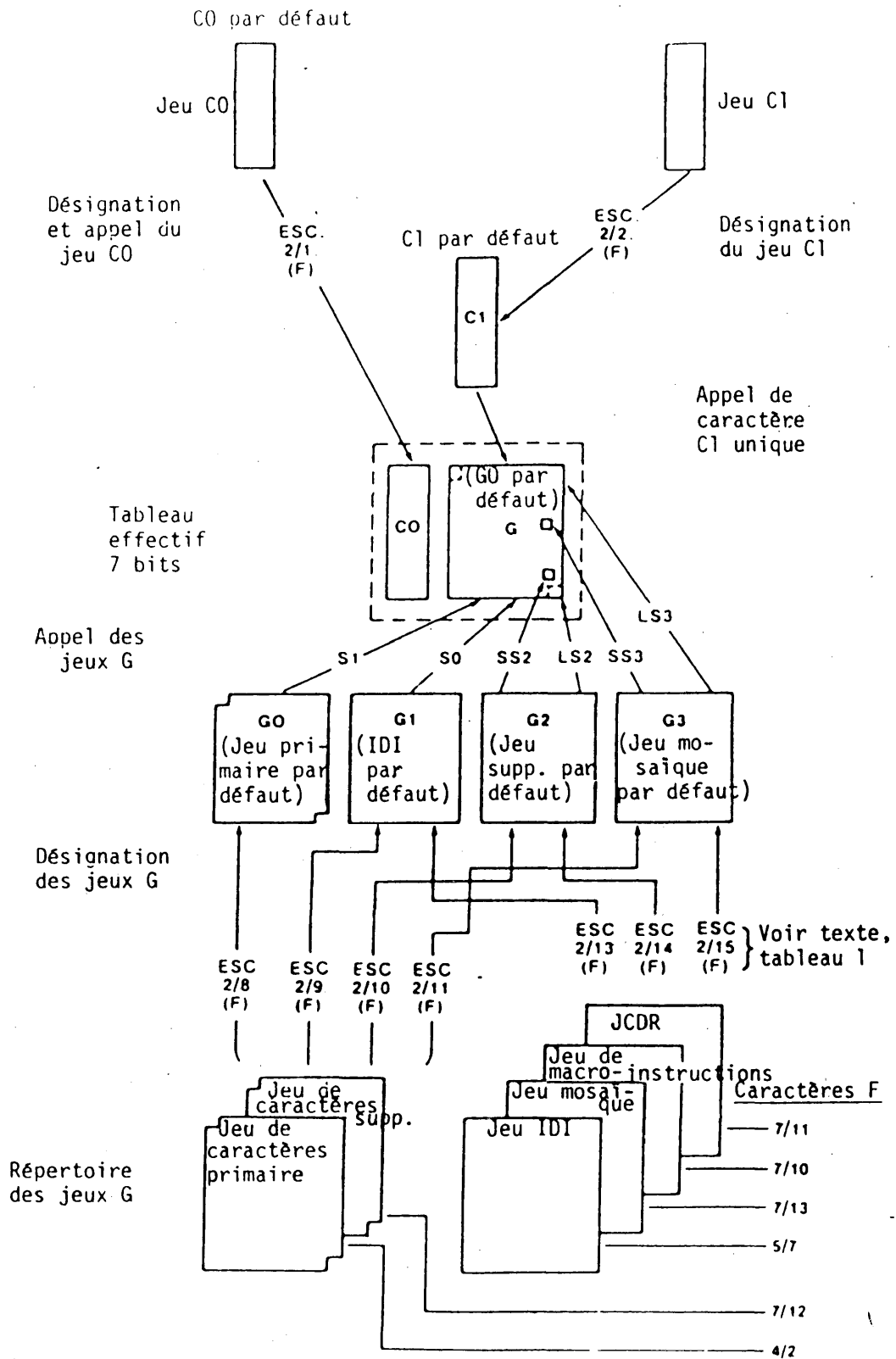


Figure 4

Extension de code à 7 éléments

Tableau 2  
Codage des fonctions d'inversion

Fonction d'inversion	Code à 7 éléments		Code à 8 éléments	Jeu G appelé
EN CODE	SI	0/15	0/15	G0 dans GL
HORS CODE	SO	0/14	0/14	G1 dans GL
INVERSION AVEC VERROUILLAGE UN-DROITE	LS1R	—	ESC 7/14*	G1 dans GR
INVERSION AVEC VERROUILLAGE DEUX	LS2	ESC 6/14	ESC 6/14	G2 dans GL
INVERSION AVEC VERROUILLAGE DEUX-DROITE	LS2R	—	ESC 7/13*	G2 dans GR
INVERSION AVEC VERROUILLAGE TROIS	LS3	ESC 6/15	ESC 6/15	G3 dans GL
INVERSION AVEC VERROUILLAGE TROIS-DROITE	LS3R	—	ESC 7/12*	G3 dans GR
INVERSION UNIQUE DEUX	SS2	1/9	1/9	G2 <sup>(sans verrouillage)</sup>
INVERSION UNIQUE TROIS	SS3	1/13	1/13	G3 <sup>(sans verrouillage)</sup>

\*LS1R, LS2R et LS3R sont aussi respectivement codés ESC 6/11, ESC 6/12 et ESC 6/13. Il est possible que ce double codage soit supprimé de la présente syntaxe de données à l'occasion d'une révision éventuelle.

Les caractères d'inversion unique, INVERSION UNIQUE DEUX (SS2) et INVERSION UNIQUE TROIS (SS3), servent respectivement à appeler sans verrouillage le jeu G2 ou G3 dans le tableau effectif. L'effet des caractères d'inversion unique se limite au caractère qui suit immédiatement, c'est-à-dire que le tableau effectif revient automatiquement à son état antérieur après l'interprétation du caractère suivant immédiatement l'inversion unique. Lorsqu'un caractère C0 suit immédiatement un caractère SS2 ou SS3 (plutôt qu'un multiplet provenant des colonnes 2 à 7), le caractère SS2 ou SS3 reste sans effet et le caractère C0 est exécuté. À noter que le jeu IDI peut faire l'objet d'une inversion unique dans le tableau effectif uniquement dans les cas où la commande IDI n'est pas suivie d'un opérateur numérique associé.

Le jeu C1 (dans le cas d'un code à 7 éléments) n'est jamais appelé avec verrouillage dans le tableau effectif. Des séquences d'échappement à deux caractères sont plutôt utilisées pour donner accès à des caractères particuliers du jeu C1. Ces séquences prennent la forme ESC Fe, où Fe représente le caractère désiré du jeu C1. Par définition, ce caractère doit comporter une combinaison de bits correspondant à la colonne 4 ou 5 du tableau effectif à 7 éléments, et il doit représenter le caractère C1 correspondant de la colonne 8 ou 9. Comme dans le cas des caractères d'inversion unique, le tableau effectif n'est pas changé par ces séquences d'échappement à deux caractères. Le tableau effectif

revient automatiquement à son état antérieur après l'exécution de la commande C1. (Bien que les commandes C1 soient toutes constituées d'un seul caractère, il est à noter que certaines commandes peuvent amorcer des opérations touchant plusieurs multipliets.)

Si l'un ou l'autre des jeux G change de désignation par suite d'une séquence d'échappement pendant qu'il se trouve dans le tableau effectif, les nouvelles interprétations de code sont appelées simultanément, c'est-à-dire que le changement ne nécessite pas une inversion avec verrouillage.

Dès l'initialisation, le jeu de caractères primaire (voir 5.1) est désigné jeu G0, et le jeu G0 est appelé par défaut dans le tableau effectif. Le jeu IDI (voir 5.3) est désigné jeu G1, le jeu de caractères supplémentaire (voir 5.2) est désigné jeu G2, et le jeu mosaïque (voir 5.4) est désigné jeu G3, par défaut dans tous les cas.

**4.3.3** Extension des codes à 8 éléments. Le codage à 8 éléments permet aussi l'extension des 256 positions de code disponibles de façon à fournir beaucoup plus d'espace pour les adresses, selon des procédures d'extension semblables à celles utilisées pour les codes à 7 éléments. Un tableau effectif à 256 positions de code est défini et illustré sur la figure 5. À nouveau, chaque combinaison de bits d'entrée est décodée en fonction du contenu de ce tableau ou utilisée pour modifier le contenu du tableau. Ce tableau se présente sous forme de 16 colonnes et de 16 rangées, les bits 1 à 4 indiquant le numéro de la rangée et les bits 5 à 8 le numéro de la colonne. Le tableau effectif renferme le jeu C0 aux colonnes 0 et 1, et le jeu C1 aux colonnes 8 et 9. Le recours aux séquences ESC Fe pour représenter les caractères C1 (voir 4.3.2) est autorisé, mais non recommandé, dans le cas des codes à 8 éléments. Les colonnes 2 à 7, appelées par convention secteur GL (G gauche) du tableau effectif, peuvent contenir l'un ou l'autre des 4 jeux appelés (G0, G1, G2 et G3). Les colonnes 10 à 15, appelées secteur GR (G droit), peuvent contenir le jeu G1, G2 ou G3. La figure 6 montre graphiquement la structure utilisée.

Le caractère SI est utilisé pour appeler avec verrouillage le jeu G0 dans GL. À noter que le jeu G0 ne peut pas être appelé dans GR. Le caractère SO est utilisé pour appeler avec verrouillage le jeu G1 dans GL. La séquence d'échappement INVERSION AVEC VERROUILLAGE UN-DROITE (LS1R) permet d'appeler avec verrouillage le jeu G1 dans GR. Les séquences d'échappement INVERSION AVEC VERROUILLAGE DEUX (LS2) et INVERSION AVEC VERROUILLAGE DEUX-DROITE (LS2R) permettent d'appeler avec verrouillage le jeu G2 dans GL et GR respectivement. Les séquences d'échappement INVERSION AVEC VERROUILLAGE TROIS (LS3) et INVERSION AVEC VERROUILLAGE TROIS-DROITE (LS3R) permettent d'appeler avec verrouillage le jeu G3 dans GL et GR respectivement. Voir le codage des fonctions d'inversion au tableau 2.

Il importe également de noter que les jeux G2 et G3 peuvent être appelés dans GL sans verrouillage, à l'aide des caractères SS2 et SS3 respectivement, comme dans le cas des codes à 7 éléments. Lorsque le multipliet suivant immédiatement le caractère SS2 ou SS3 provient des colonnes 10 à 15, le bit b8 reste sans effet.

Dès l'initialisation, les jeux primaire, IDI, supplémentaire et mosaïque sont respectivement désignés jeux G0, G1, G2 et G3, et G0 est appelé par défaut dans GL, comme avec les codes à 7 éléments. En outre, G1 est appelé dans GR.



**4.4 ESPACE et OBLITÉRATION.** La fonction ESPACE correspond à un champ de caractère vide pouvant faire l'objet des mêmes attributs que les caractères alphanumériques. Sa position est 2/0 dans le tableau effectif à 7 et à 8 éléments lorsqu'un jeu G à 94 positions de code est appelé.

La fonction OBLITÉRATION est principalement utilisée pour effacer ou oblitérer des caractères erronés ou non désirés sur une bande perforée. Sa position est 7/15 dans le tableau effectif à 7 et à 8 éléments lorsqu'un jeu G à 94 positions de code est appelé. Dans le contexte de la présente norme, la fonction OBLITÉRATION est exécutée comme s'il s'agissait d'une opération vide.

				COLONNE															
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	RANGE																
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1
0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

				COLONNE															
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5  
Tableau effectif à 8 éléments

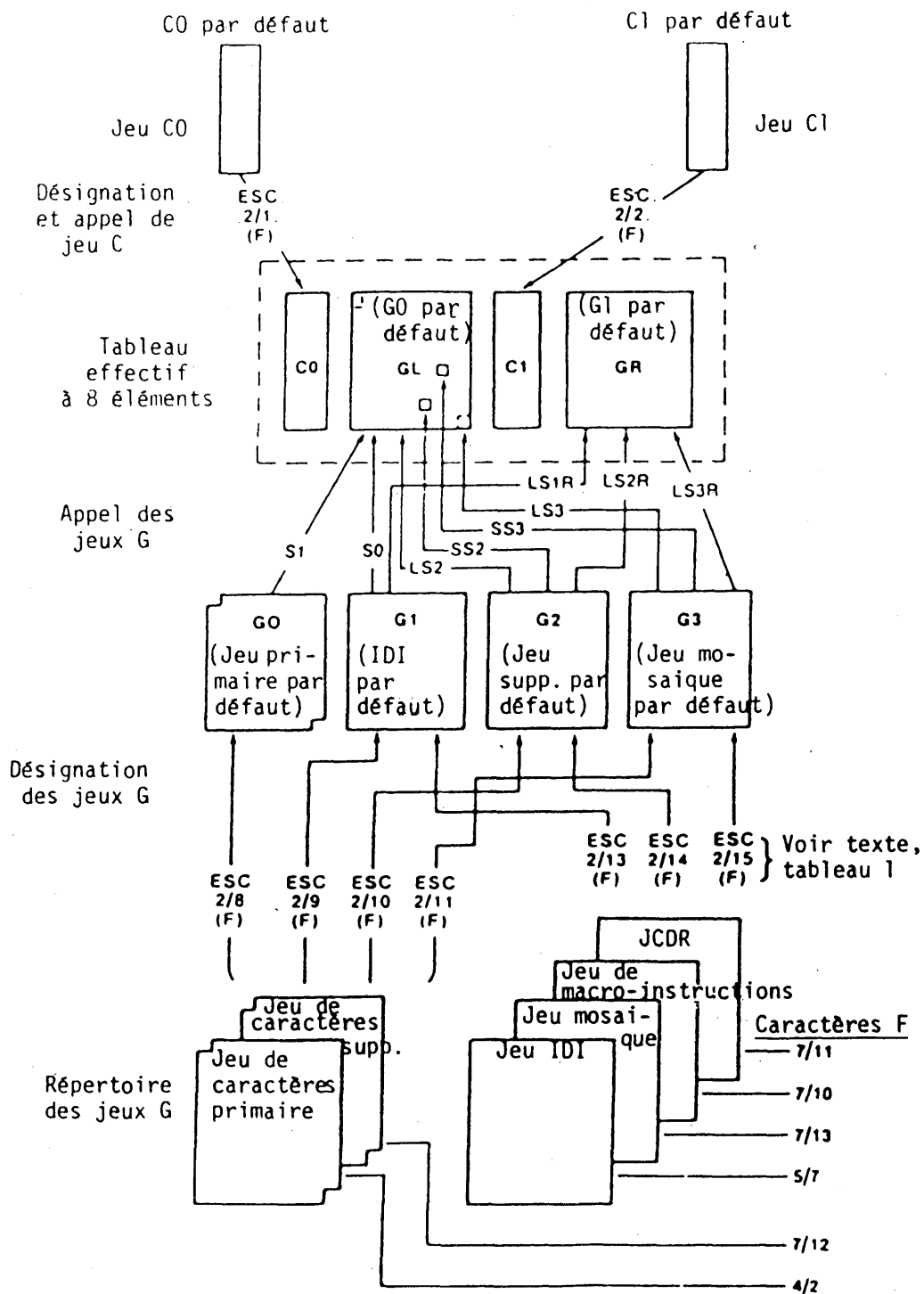


Figure 6

Extension des codes à 8 éléments

## 5. Codage des jeux G

**5.1 Jeu de caractères primaire.** Le jeu de caractères primaire est constitué des 94 caractères alphabétiques latins, ainsi que des chiffres, signes de ponctuation et symboles illustrés sur la figure 7. Ce jeu est identique à celui décrit dans l'Avis V.3 du CCITT, dont la version nationale a été adoptée par le Canada et les États-Unis d'Amérique. Les configurations (fontes) particulières choisies pour les caractères sont liées à la matérialisation et limitées seulement par le champ de caractère spécifié pour chaque taille, compte tenu d'une résolution d'affichage donnée. La lisibilité des caractères n'est pas garantie pour toutes les tailles, couleurs et résolutions d'affichage. À la réception d'un caractère quelconque du jeu primaire, le curseur avance automatiquement (voir 5.3.2.3.4).

La séquence utilisée pour désigner le jeu de caractères primaire est ESC I 4/2, où I désigne 2/8, 2/9, 2/10 ou 2/11, soit respectivement G0, G1, G2 ou G3 (voir 4.3).

					10	11	12	13	14	15	
					b7	0	0	1	1	1	1
					b6	1	1	0	0	1	1
					b5	0	1	0	1	0	1
					COLONNE	2	3	4	5	6	7
b4	b3	b2	b1	RANGÉE							
0	0	0	0	0		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	l	
1	1	0	0	12	,	<	L	\	l		
1	1	0	1	13	-	=	M		m		
1	1	1	0	14	.	>	N	^	n	~	
1	1	1	1	15	/	?	O	_	o		

Figure 7

Jeu de caractère primaire

**5.2 Jeu de caractères supplémentaire.** Le jeu de caractères supplémentaire des accents, signes diacritiques et caractères spéciaux des alphabets d'origine latine est illustré sur la figure 8. Ce tableau se fonde sur l'Avis S.100-1980 du CCITT et comprend aussi certains caractères additionnels proposés par le CCIR, l'ISO et d'autres organismes. Les configurations (fontes) particulières choisies pour ces caractères sont liées à la matérialisation et limitées seulement par le champ de caractère spécifié pour chaque taille, compte tenu d'une résolution d'affichage donnée. Les 16 accents et symboles de la colonne 4 du tableau sont traités différemment de tous les autres caractères du fait qu'ils sont sans espacement. Ainsi, contrairement à la normale, la réception d'un de ces caractères ne fait pas automatiquement avancer le curseur (voir 5.3.2.3.4).

Le codage d'un caractère accentué s'obtient par l'adjonction d'un accent sans espacement provenant du jeu supplémentaire à la lettre visée provenant du jeu primaire. Seules certaines combinaisons de caractères sans espacement du jeu supplémentaire peuvent être associées aux caractères du jeu primaire afin de former des caractères du répertoire graphique (voir 7.2). Le codage d'un caractère composé requiert habituellement trois multiplets. Si l'on suppose par exemple qu'un codage à 7 éléments est utilisé, que le jeu primaire est désigné jeu G0 par défaut et est appelé dans le tableau effectif, et que le jeu supplémentaire est désigné G2, le codage correspondant à ë (e avec tréma) se fait comme suit. Un caractère SS2 (position 1/9 du jeu C0) amorce la séquence en appelant un caractère unique du tableau de code G2. Vient ensuite le symbole du tréma (``), suivi du caractère primaire. De cette façon, la lettre ë est codée SS2"e, soit les trois caractères aux positions 1/9, 4/8 et 6/5 du tableau de code. Dans le cas d'un code à 8 éléments, le codage peut s'effectuer de la même façon ou, si le jeu primaire est appelé dans G1 et que le jeu supplémentaire est appelé dans GR, la lettre ë peut être codée "e, soit les deux caractères aux positions 12/8 et 6/5 du tableau de code.

La séquence utilisée pour désigner le jeu de caractères supplémentaire est ESC I 7/12, où I désigne 2/8, 2/9, 2/10 ou 2/11, soit respectivement G0, G1, G2 ou G3 (voir 4.3).

					10	11	12	13	14	15	
					b7	0	0	1	1	1	1
					b6	1	1	0	0	1	1
					b5	0	1	0	1	0	1
					COLONNE	2	3	4	5	6	7
b4	b3	b2	b1	RANGÉE							
0	0	0	0	0		0	—	—	Ω	℔	
0	0	0	1	1		ı	±	`	ı	Æ	æ
0	0	1	0	2		¢	²	˙	®	Ð	đ
0	0	1	1	3		£	³	˘	©	ä	δ
0	1	0	0	4		§	×	~	T.M.	ℋ	ℎ
0	1	0	1	5		¥	μ	—	♪	⌘	ı
0	1	1	0	6		#	¶	˘	☐	ıı	ıı
0	1	1	1	7		§	˙	˙	☐	Ł	ł
1	0	0	0	8		¤	÷	˙˙	☐	Ł	ł
1	0	0	1	9		‘	’	/	☐	Ø	ø
1	0	1	0	10		“	”	˙	☐	Œ	œ
1	0	1	1	11		«	»	˘	☐	Œ	œ
1	1	0	0	12		←	¼	☐	⅛	þ	þ
1	1	0	1	13		↑	½	”	⅜	ƒ	ƒ
1	1	1	0	14		→	¾	˘	⅝	ŋ	ŋ
1	1	1	1	15		↓	ı	˘	⅞	˘n	

Note: La colonne 4 (12) regroupe des caractères sans espacement. Par ailleurs, les rectangles entourant les caractères en 4/12, 5/6 à 5/11, et 6/5 sont tracés aux fins d'illustration; ils ne font pas partie intégrante des symboles graphiques.

Figure 8

Jeu de caractères supplémentaire

### 5.3 Jeu d'instructions de description de l'image (IDI)

#### 5.3.1 Généralités

5.3.1.1 Le jeu d'instructions de description de l'image (IDI), illustré sur la figure 9, comprend 6 primitives graphiques géométriques (POINT, LIGNE, ARC, RECTANGLE, POLYGONE et INCRÉMENTATION), chacune de ces primitives comportant quatre variantes; le jeu comprend aussi 8 codes de commande (INITIALISATION, DOMAINE, TEXTE, TEXTURE, DÉTERMINATION COULEUR, ATTENTE, SÉLECTION COULEUR et CLIGNOTEMENT), ainsi que 64 positions de caractère pour données numériques (correspondant à un champ de données de 6 bits dans chaque multiplet d'information). Le jeu IDI se différencie essentiellement des jeux de caractères alphanumériques par le fait qu'il ne se compose pas de configurations prédéfinies, une par caractère, mais qu'il regroupe plutôt des fonctions de dessin exécutables produisant une image qui n'est pas nécessairement restreinte à un seul champ de caractère.

La séquence utilisée pour désigner le jeu IDI est ESC I 5/7, la lettre I représentant 2/9 ou 2/13 pour indiquer G1, 2/10 ou 2/14 pour indiquer G2, ou 2/11 ou 2/15 pour indiquer G3 (voir 4.3).

					10	11	12	13	14	15									
					b7	0	0	1	1	1	1								
					b6	1	1	0	0	1	1								
					b5	0	1	0	1	0	1								
					COLONNE	2	3	4	5	6	7								
b4	b3	b2	b1	RANGÉE	<table><tr><td>COMMANDE</td><td>RECTANGLE</td></tr><tr><td>POINT</td><td>POLYGONE</td></tr><tr><td>LIGNE</td><td>INCRÉMENTATION</td></tr><tr><td>ARC</td><td>COMMANDE</td></tr></table> <div>DONNÉES NUMÉRIQUES</div>							COMMANDE	RECTANGLE	POINT	POLYGONE	LIGNE	INCRÉMENTATION	ARC	COMMANDE
COMMANDE	RECTANGLE																		
POINT	POLYGONE																		
LIGNE	INCRÉMENTATION																		
ARC	COMMANDE																		
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															

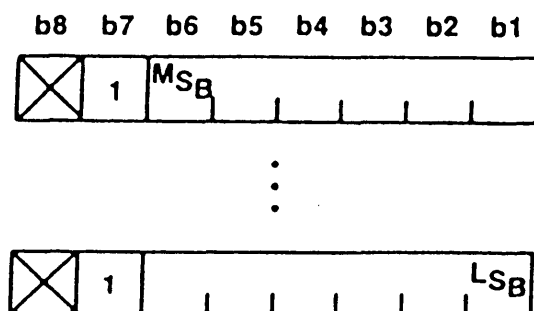
Figure 9

Jeu IDI général



Une IDI se compose d'un code d'opération, qui doit correspondre à l'une des quatre formes des six primitives graphiques ou constituer l'un des huit codes de commande, suivi d'aucun, d'un ou de plusieurs opérandes, comportant chacun un ou plusieurs multiplats de données numériques. Le code d'opération se distingue toujours de l'opérande par la valeur du bit 7: si ce bit est à zéro, il s'agit d'un code d'opération et s'il est à un, de données numériques (c'est-à-dire d'un opérande). Une séquence IDI se termine par un code d'opération introduisant la séquence IDI suivante, ou par tout autre code de la couche présentation ne provenant pas de la section des données numériques du même jeu IDI. Les caractères de commande de transmission (0/1-0/6, 1/0, 1/5-1/7), les caractères de commande de périphérique (1/1-1/4), et le caractère NUL (0/0) n'ont aucune incidence sur la couche présentation et ne peuvent donc pas mettre fin à des séquences IDI (voir 6.1.4, 6.1.5 et 6.1.6.1). L'appel d'une macro-instruction par accès au tableau effectif ou par inversion unique ne suffit pas pour mettre fin à une IDI; l'IDI peut en effet se poursuivre dans les données d'opérande que contient le macro-code. Il existe quatre types d'opérandes: à format fixe, en chaîne, à valeur simple et à valeurs multiples.

Les opérandes à format fixe sont constitués d'un ou de plusieurs multiplats de données numériques dont la longueur et l'interprétation dépendent du code d'opération auquel ils sont associés. Les chaînes opérandes sont de longueur indéterminée, c'est-à-dire qu'elles peuvent comporter n'importe quel nombre de multiplats de données numériques. Leur interprétation dépend également du code d'opération auquel elles sont associées, mais leur décodage se fait toujours de gauche à droite, c'est-à-dire de b6 à b1. Les opérandes à valeur simple sont constitués d'un à quatre multiplats de données numériques, selon la commande DOMAINE décrite en 5.3.2.2. Ils sont interprétés comme des nombres entiers (nombres ordinaux) non affectés d'un signe et composés de la séquence des bits concaténés prélevés consécutivement (du bit de poids fort, ou b6, au bit de poids faible, ou b1) des multiplats de données numériques, comme le montre la figure 10.



MSB - BIT DE POIDS FORT DE L'OPÉRANDE

LSG - BIT DE POIDS FAIBLE DE L'OPERANDE

MULTIPLIET DE POIDS FORT TRANSMIS EN PREMIER

Figure 10

Format à valeur simple

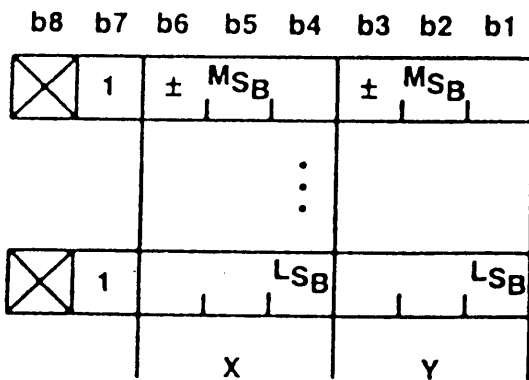
Les opérandes à valeurs multiples sont constitués d'un à huit multipléts de données numériques, selon la commande DOMAINE. Ces opérandes servent à spécifier l'information de coordonnée (lorsqu'ils sont associés à des primitives graphiques) ou l'information de couleur (lorsqu'ils sont associés à la commande DÉTERMINATION COULEUR).

Les coordonnées sont établies selon un système unitaire cartésien de numérotation, les positions étant spécifiées par des fractions de la plage de 0 (inclusivement) à 1 (exclusivement).

Les données des coordonnées définies par les opérandes IDI peuvent s'interpréter comme les coordonnées absolues d'une position de dessin logique à l'intérieur de l'écran unitaire ou comme des déplacements par rapport à la position de dessin antérieure selon le contexte établi par le code d'opération particulier. Cette position de dessin sert ensuite à l'exécution des primitives géométriques, décrites en 5.3.3.

La figure 11 illustre la représentation des coordonnées dans un opérande à valeurs multiples.

#### MODE BIDIMENSIONNEL



#### MODE TRIDIMENSIONNEL

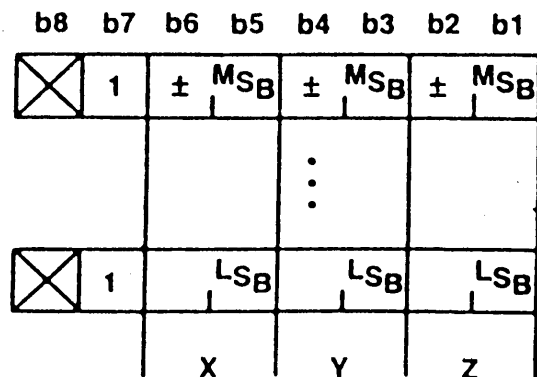


Figure 11  
Format à valeurs multiples

Tous les opérandes de coordonnées sont interprétés comme des nombres affectés d'un signe et exprimés sous forme de complément à deux, c'est-à-dire comme des décimales binaires dont le bit de poids fort (MSB) représente le chiffre situé juste à la droite du point décimal. La précision à laquelle doivent être maintenus l'emplacement du curseur et la position de dessin est liée à la matérialisation, compte tenu de la résolution physique. Lorsqu'une spécification de coordonnées ou qu'une opération de dessin amènerait la position de dessin ou une partie quelconque du dessin résultant à l'extérieur de l'écran unitaire, l'IDI est considérée comme erronée. Le traitement de la condition d'erreur est lié à la matérialisation. À titre d'exemple, cette IDI peut être rejetée (c'est-à-dire traitée comme une opération vide) ou exécutée et limitée à l'intérieur de l'écran unitaire.

Lorsqu'il est associé à la commande DÉTERMINATION COULEUR décrite en 5.3.2.5, l'opérande à valeurs multiples spécifie une valeur non affectée d'un signe dans le système des couleurs VRB (vert-rouge-bleu). La figure 12 illustre la représentation des couleurs dans un opérande à valeurs multiples.

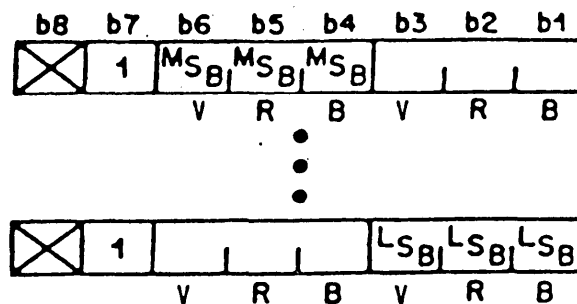


Figure 12

Format de valeur de couleur

Chaque multiplet renferme deux triplets. Dans chacun de ces triplets, un bit est associé à chacune des trois couleurs primaires. Ces couleurs sont spécifiées dans l'ordre vert, rouge, bleu, c'est-à-dire dans l'ordre de luminance décroissante. La valeur de couleur complète de chaque primaire est donnée par la concaténation des bits extraits de chacun des triplets en commençant par le bit de poids fort indiqué et en continuant de gauche à droite jusqu'au bit de poids faible indiqué. La valeur de couleur ainsi obtenue correspond à une fraction binaire dans laquelle le bit de poids fort représente le chiffre situé juste à la droite du point décimal.

Le tableau 3 indique les types d'opérandes utilisés pour chacun des codes d'opération.

Tableau 3  
Types d'opérandes

Code d'opération	Opérande
INITIALISATION	Fixe
DOMAINE	Fixe/valeurs multiples
TEXTE	Fixe/valeurs multiples
TEXTURE	Fixe/valeurs multiples
DÉTERMINATION COULEUR	Valeurs multiples
ATTENTE	Fixe
SÉLECTION COULEUR	Valeur simple
CLIGNOTEMENT	Fixe/valeur simple
POINT	Valeurs multiples
LIGNE	Valeurs multiples
ARC	Valeurs multiples
RECTANGLE	Valeurs multiples
POLYGONE	Valeurs multiples
CHAMP	Valeurs multiples
INCRÉMENTATION POINT	Fixe/chaîne
INCRÉMENTATION LIGNE	Valeurs multiples/chaîne
INCRÉMENTATION POLYGONE (REPLI)	Valeurs multiples/chaîne

5.3.1.2 Les fonctions des codes d'opération sont résumées ci-dessous.

(1) POINT: établit la position de dessin n'importe où sur l'écran unitaire et affiche facultativement un point.

(2) LIGNE: trace une ligne entre deux extrémités définies.

(3) ARC: trace un arc de cercle à partir de trois points: les extrémités de l'arc et un point situé sur l'arc. Il est possible de joindre les extrémités de l'arc par une corde et de remplir l'aire ainsi définie. Si plus de points sont donnés, ils définissent un arc plus complexe, c'est-à-dire un tracé curviligne défini par une fonction irrégulière. Le cercle est défini comme un arc dont les extrémités coïncident et dont le point intermédiaire (par rapport aux extrémités) établit le diamètre.

(4) RECTANGLE: trace un périmètre rectangulaire ou remplit une aire de longueur et de largeur spécifiées.

(5) POLYGONE: trace un périmètre polygonal ou remplit l'aire circonscrite à partir d'une série de sommets spécifiés.

(6) INCRÉMENTATION: trace un point, une ligne ou un polygone par incrémentation.

(7) COMMANDE: assure la gestion des modes de commande de dessin. Une des principales fonctions consiste à déterminer une valeur ou la couleur d'un objet.

La figure 13 montre la configuration détaillée du jeu IDI et identifie chaque forme des primitives géométriques et codes de commande.

					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
					COLONNE	2	3	4	5	6	7
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	RANGÉE							
0	0	0	0	0	INITIALISATION	RECT. (PÉRI.)	DONNÉES NUMÉRIQUES				
0	0	0	1	1	DOMAINE	RECT. (REMP.)					
0	0	1	0	2	TEXTE	DÉTERM. RECT. (PÉRI.)					
0	0	1	1	3	TEXTURE	DÉTERM. RECT. (REMP.)					
0	1	0	0	4	DÉTERM. POINT (ABS.)	POLY. (PÉRI.)					
0	1	0	1	5	DÉTERM. POINT (REL.)	POLY. (REMP.)					
0	1	1	0	6	POINT (ABS.)	DÉTERM. POLY. (PÉRI.)					
0	1	1	1	7	POINT (REL.)	DÉTERM. POLY. (REMP.)					
1	0	0	0	8	LIGNE (ABS.)	CHAMP					
1	0	0	1	9	LIGNE (REL.)	INCRÉ. POINT					
1	0	1	0	10	DÉTERM. LIGNE (ABS.)	INCRÉ. LIGNE					
1	0	1	1	11	DÉTERM. LIGNE (REL.)	INCRÉ. POLY. (REMP.)					
1	1	0	0	12	ARC (PÉRI.)	DÉTERM. COULEUR					
1	1	0	1	13	ARC (REMP.)	ATTENTE					
1	1	1	0	14	DÉTERM. ARC (PÉRI.)	SELECT COULEUR					
1	1	1	1	15	DÉTERM. ARC (REMP.)	CLIGNOTEMENT					

Figure 13

Jeu IDI

## 5.3.2 Fonctions de commande d'attribut

**5.3.2.1 Généralités.** Les codes d'opération décrits en 5.3.2.2 à 5.3.2.9 commandent les attributs et les paramètres d'affichage.

### 5.3.2.2 DOMAINE

**5.3.2.2.1 Format de la commande.** La commande DOMAINE sert à établir la précision des opérands à valeur simple et à valeurs multiples, le mode dimensionnel des spécifications de coordonnées et la taille du pel logique. (Voir la figure 14.) Une fois déterminés, ces paramètres ne changent plus à moins d'être visés par la commande INITIALISATION, une autre commande DOMAINE ou le code de commande NSR décrit en 6.1.6.5. Le code d'opération DOMAINE constitue un opérande à format fixe d'un multiplet, et il est suivi d'un opérande à valeurs multiples dont les interprétations sont données ci-dessous.

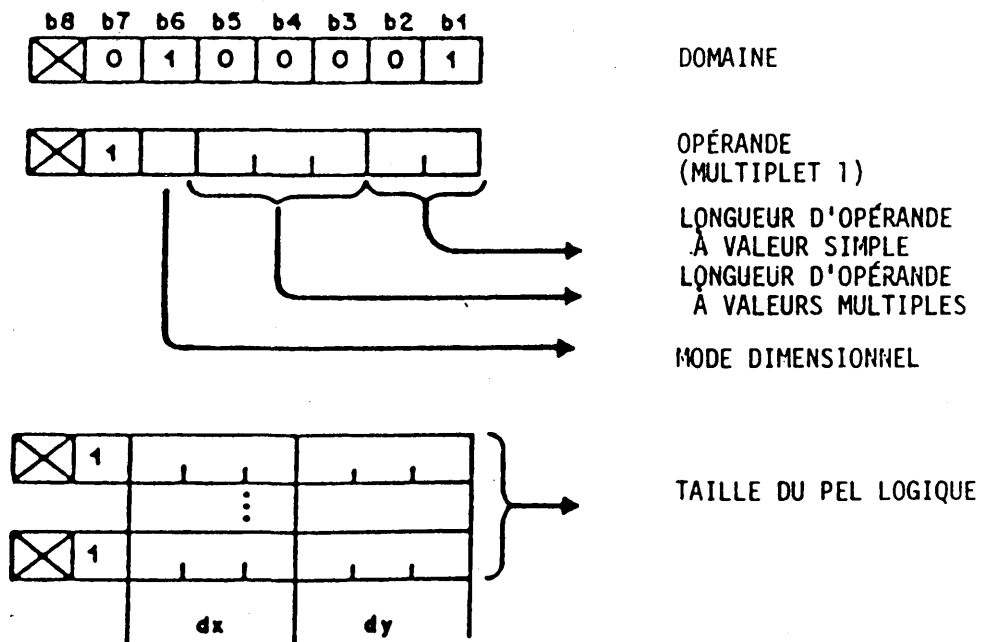


Figure 14  
Domaine

5.3.2.2.2 Longueur d'opérande à valeur simple. Les bits b2 et b1 du multiplet 1 déterminent la longueur, c'est-à-dire le nombre de multi-plets, qui doit être utilisée pour les opérandes à valeur simple, de la façon indiquée au tableau 4. La longueur par défaut est un multiplet.

Tableau 4  
Longueur d'opérande à valeur simple

b2	b1	Nombre de multi-plets
0	0	1 (par défaut)
0	1	2
1	0	3
1	1	4

5.3.2.2.3 Longueur d'opérande à valeurs multiples. Les bits b5, b4 et b3 du multiplet 1 déterminent la longueur, c'est-à-dire le nombre de multi-plets, qui doit être utilisée pour les opérandes à valeurs multiples, de la façon indiquée au tableau 5. La longueur par défaut est trois multi-plets.

Tableau 5  
Longueur d'opérande à valeurs multiples

b5	b4	b3	Nombre de multi-plets
0	0	0	1
0	0	1	2
0	1	0	3 (par défaut)
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

5.3.2.2.4 Mode dimensionnel. Le bit 6 du multiplet 1 détermine le mode dimensionnel de la spécification de coordonnées. Un 0 indique le mode bidimensionnel (X,Y), c'est-à-dire le mode par défaut. Un 1 indique le mode tridimensionnel (X,Y,Z). Lorsque des coordonnées tridimensionnelles sont reçues, la coordonnée Z doit rester sans effet, afin de projeter l'image dans le plan bidimensionnel (X,Y). La définition complète du mode tridimensionnel sera normalisée plus tard.



**5.3.2.2.5 Longueur d'opérande.** Lorsqu'un opérande suivant un code d'opération n'atteint pas la longueur spécifiée antérieurement par la commande DOMAINE (ou la longueur implicite dans le cas du format fixe), des zéros non significatifs sont ajoutés dans le processus de présentation à la réception, sauf indication contraire dans la définition de la commande. Lorsqu'un opérande suivant un code d'opération dépasse la longueur spécifiée antérieurement par la commande DOMAINE (ou la longueur implicite), le code d'opération est exécuté à nouveau et les données numériques subséquentes sont utilisées comme opérandes, sauf indication contraire dans la présente syntaxe de données.

**5.3.2.2.6 Pel logique.** Les coordonnées qui suivent le multiplet 1 de l'opérande sont interprétées comme la largeur (dx) et la hauteur (dy) du pel logique, soit un rectangle dont l'orientation est fixe par rapport au système de coordonnées cartésiennes. Cet opérande à valeurs multiples spécifie la taille du pel logique à utiliser avec les IDI POINT, LIGNE, ARC, RECTANGLE, POLYGONE et INCRÉMENTATION, avec la commande DÉBUT SOULIGNEMENT, les formes mosaïques distinctes, les textures de ligne et les motifs de texture, mais pas pour l'affichage des caractères alphanumériques (y compris le soulignement et le soulignement sans espacement). Cette spécification est obtenue par la définition des opérations de dessin de façon qu'elles s'appliquent à tous les pixels visés par une partie quelconque du pel logique transposé sur l'écran d'affichage. Le pel logique est donc toujours transposé sur au moins un pixel, et parfois sur un grand nombre. À noter que si la largeur et la hauteur du pel logique sont toutes deux ramenées à zéro, le pel logique est réduit à une position de dessin infinitésimale. La taille par défaut du pel logique est  $dx = 0$ ,  $dy = 0$ , l'origine se trouvant au coin inférieur gauche.

Une primitive de dessin se définit par un algorithme lié à la matérialisation, qui décrit le mieux possible un trajet géométrique précis pour tous les déplacements, y compris les déplacements nuls. Une LIGNE, par exemple, est le lieu des points suivant un algorithme de droite entre deux coordonnées spécifiées. Les éléments physiques de l'image (pixels) par lesquels passe le point infinitésimal sont de ce fait excités. La spécification du pel logique détermine avec précision la taille du point dont le rôle se compare à celui d'un "pinceau" qui excite les pixels voulus sur son trajet géométrique, produisant une ligne caractérisée par une certaine largeur. Voir la figure 15, qui illustre l'application du pel logique à une ligne, un arc et un point.

Le point d'alignement géométrique de la position de dessin du pel logique est:

- (1) le coin inférieur gauche si les valeurs dx et dy sont positives;
- (2) le coin inférieur droit si la valeur dx est négative et la valeur dy positive;
- (3) le coin supérieur gauche si la valeur dx est positive et la valeur dy négative; et
- (4) le coin supérieur droit si les valeurs dx et dy sont négatives.

À noter que la nouvelle longueur des opérandes à valeurs multiples, déterminée par le multiplet 1, s'applique à l'opérande à valeurs multiples associé à la commande DOMAINE en question pour établir la taille du pel logique.

Les multiplats de données numériques suivant les multiplats de taille du pel logique seront normalisés plus tard et doivent donc rester sans effet. La taille du pel logique doit rester inchangée lorsque l'opérande de taille du pel logique est omis.

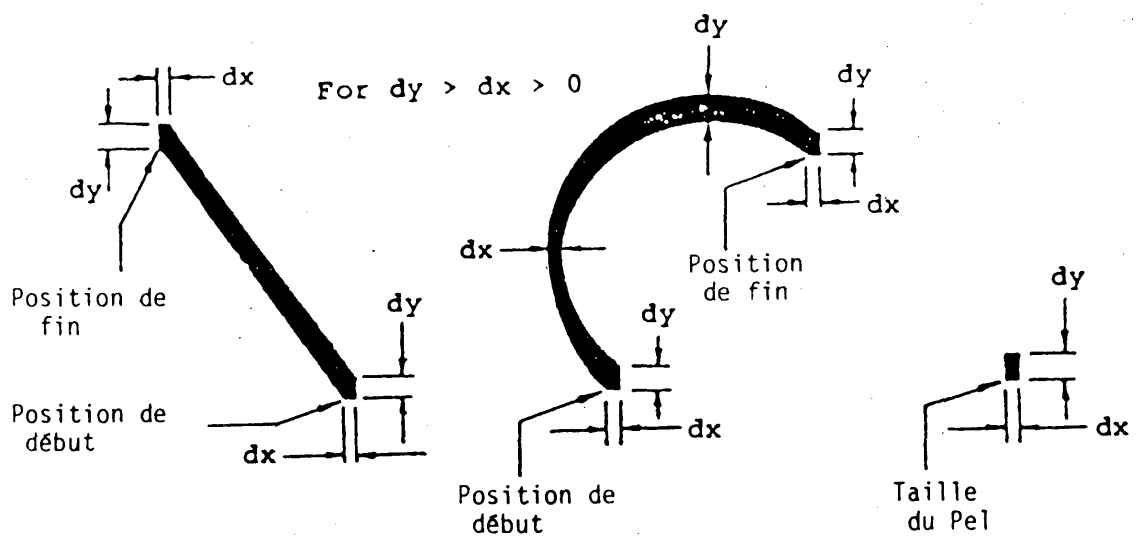


Figure 15

Application d'un pel logique à une ligne, un arc et un point  
(Incidence de la taille du pel logique sur la largeur de trait)

### 5.3.2.3 TEXTE

5.3.2.3.1 Format de la commande. Cette commande permet de modifier les paramètres qui décrivent la façon dont sont présentés les caractères alphanumériques, mosaïques et JCDR subséquents. Le code d'opération TEXTE occupe un opérande à format fixe de deux multiplets, et il est suivi d'un opérande à valeurs multiples dont les interprétations sont données ci-dessous. (Voir la figure 16.)

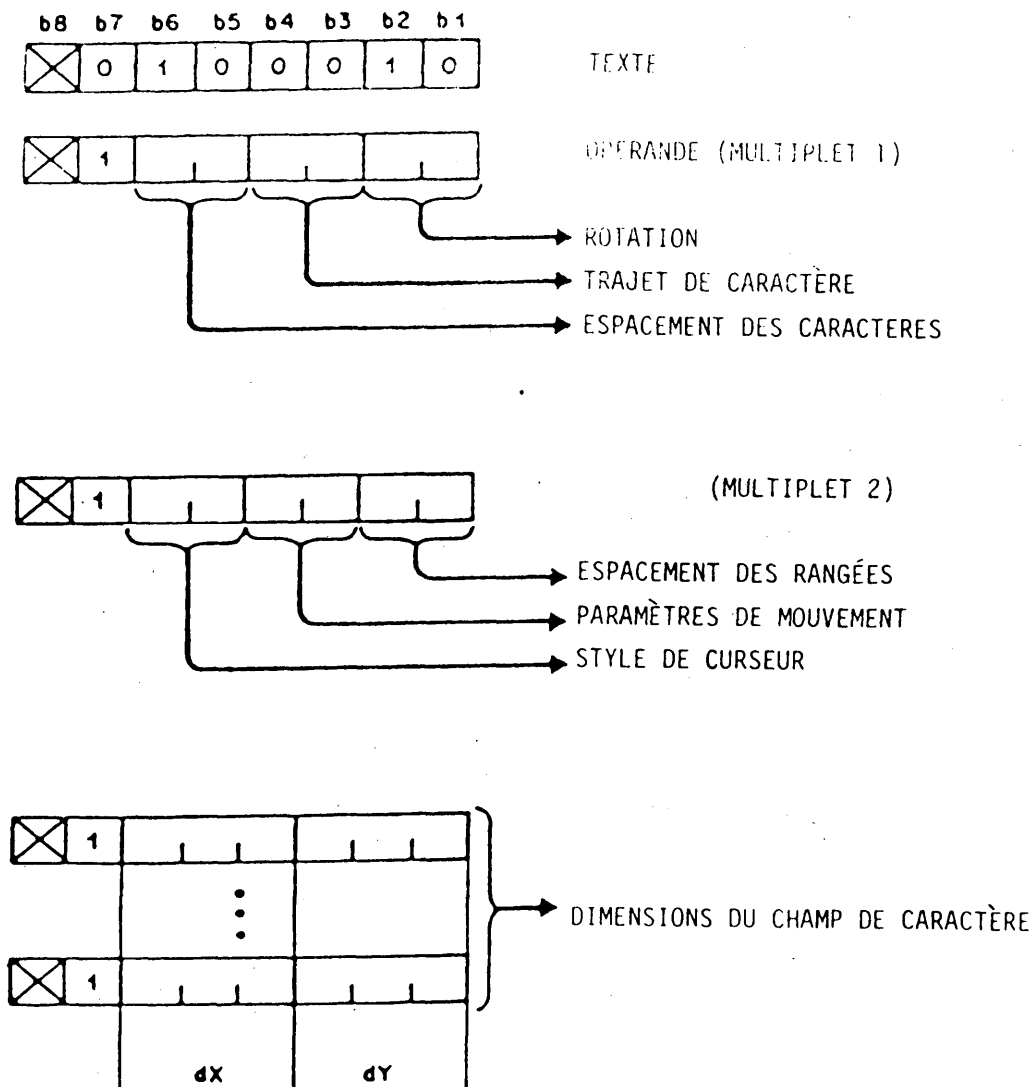


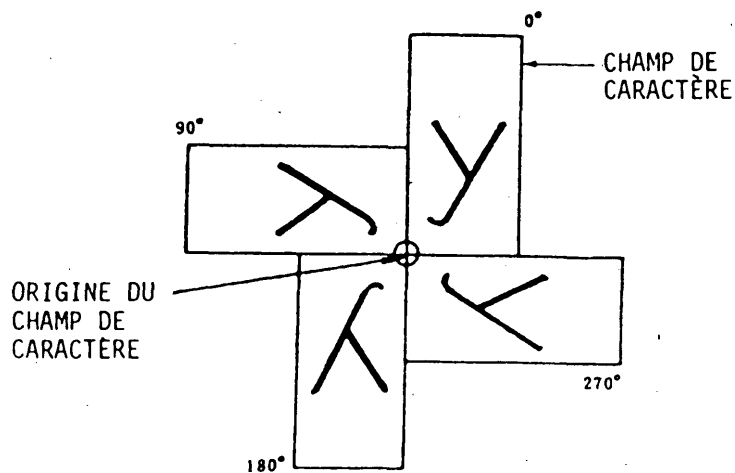
Figure 16  
Texte

**5.3.2.3.2 Rotation de caractère.** Les bits b2 et b1 du multiplet 1 servent à spécifier la rotation de caractère, de la façon indiquée au tableau 6.

**Tableau 6**  
**Rotation de caractère**

b2	b1	Degrés de rotation
0	0	0 (par défaut)
0	1	90
1	0	180
1	1	270

La rotation fait se déplacer le champ de caractère et le curseur dans le sens antihoraire par rapport à l'origine du champ de caractère. La rotation se mesure par rapport à une horizontale tracée à l'intérieur de l'écran unitaire, et elle n'est pas liée au trajet de caractère. L'origine du champ de caractère est le coin inférieur gauche du champ de caractère dans la position par défaut de 0 degré, quel que soit le signe des dimensions dx et dy du champ (voir figure 17). Tous les caractères alphanumériques (y compris les signes diacritiques et les soulignés), le JCDR, le jeu mosaïque et les caractères mosaïques distincts, ainsi que les soulignés produits en mode soulignement (voir 6.2.7.15), sont touchés par le mouvement de rotation, de sorte que la position relative des images reste inchangée à l'intérieur du champ de caractère.



**Figure 17**  
**Rotation de caractère**

**5.3.2.3.3 Sens du trajet de caractère.** Les bits b4 et b3 du multiplet 1 déterminent le sens du trajet de caractère, c'est-à-dire le sens dans lequel le curseur avance automatiquement après l'application d'un caractère. Le tableau 7 indique les quatre trajets de caractère possibles. Le trajet de caractère se définit par rapport à une horizontale tracée à l'intérieur de l'écran unitaire, et il n'est pas lié à la rotation de caractère. Le trajet de caractère par défaut est vers la droite.

**Tableau 7**  
**Trajet de caractère**

b4	b3	Mouvement du curseur
0	0	Vers la droite (par défaut)
0	1	Vers la gauche
1	0	Vers le haut
1	1	Vers le bas

**5.3.2.3.4 Espacement des caractères.** Les bits b6 et b5 du multiplet 1 déterminent la distance sur laquelle le curseur se déplace après l'affichage d'un caractère ou la réception du caractère ESPACE, APB (retour arrière) ou APF (tabulation horizontale). La distance sur laquelle se déplace le curseur est exprimée en multiples de la largeur (dx) ou de la hauteur (dy) du champ de caractère, parallèlement au trajet de caractère et selon les spécifications de ce trajet et de la rotation de caractère. Cette distance, appelée espacement des caractères, est définie au tableau 8.

**Tableau 8**  
**Espacement des caractères**

b6	b5	Espacement des caractères
0	0	1 (par défaut)
0	1	5/4
1	0	3/2
1	1	Espacement proportionnel

Les trois espacements fixes des caractères (1, 5/4 et 3/2, compte tenu de la résolution physique) sont interprétés comme des fonctions multiplicatrices de la dimension du champ de caractère actuel située parallèlement au trajet de

caractère et sont appliqués aux mouvements du curseur. En mode d'espacement proportionnel, l'espacement des caractères est une variable pouvant être fonction de la largeur du motif actuel appliqué ainsi que de la taille actuelle des caractères et de leur style de fonte. L'algorithme d'espacement proportionnel est lié à la matérialisation. Chaque caractère doit cependant entrer complètement dans l'aire définie par le champ de caractère actuel (voir 5.3.2.3.9). Cela signifie que le nombre exact de caractères par ligne est inconnu en mode d'espacement proportionnel, mais qu'il est au moins égal au nombre minimal de caractères permis par les dimensions du champ de caractère actuel. Afin de garantir l'affichage d'un texte à espacement proportionnel dans un champ actif (voir 5.3.3.6.2) pour toutes les matérialisations, sans risques de renouement (voir 6.2.7.11 et 5.3.2.3.6) ni de défilement (voir 6.2.7.13) non désiré, il est à noter que:

- (1) le champ doit être suffisamment grand pour contenir le texte en entier, comme si celui-ci n'était pas à espacement proportionnel, ou
- (2) le nombre et la taille des caractères doivent être suffisamment réduits pour qu'il y ait adaptation au champ, comme si le texte n'était pas à espacement proportionnel.

La largeur du champ de caractère ne change pas. Ainsi, lorsqu'un caractère est affiché à la fin d'une ligne en mode de couleur 2, la couleur de fond apparaît sur toute la largeur du champ de caractère. L'espacement des caractères par défaut est un espacement fixe de 1, le champ de caractère actuel touchant le champ de caractère précédent.

**5.3.2.3.5 Espacement des rangées.** Les bits b2 et b1 du multiplet 2 déterminent l'espacement des rangées de caractères, ce qui définit la position relative du curseur lorsque celui-ci passe à une nouvelle ligne perpendiculairement (-90 degrés) au trajet de caractère, de la façon automatique décrite ci-dessous ou par l'action du caractère APD (interligne) ou APU (tabulation verticale), définie en 6.1.2. Le tableau 9 indique quatre espacements des rangées (1, 5/4, 3/2 et 2, compte tenu de la résolution physique), qui sont interprétés comme des multiples de la largeur (dx) ou de la hauteur (dy) du champ de caractère, perpendiculairement au trajet de caractère et selon les spécifications de ce trajet et de la rotation de caractère. L'espacement des rangées par défaut est de 1, le champ de caractère de la rangée actuelle touchant alors le champ de caractère de la rangée précédente.

Tableau 9

Espacement des rangées

b2	b1	Espacement
0	0	1 (par défaut)
0	1	5/4
1	0	3/2
1	1	2

**5.3.2.3.6 Séquence automatique APR APD.** Lorsque, avec un espacement fixe ou proportionnel des caractères, le déplacement du curseur (par l'emploi d'un caractère de commande de mise en page ou l'affichage d'un caractère) entraînerait une partie quelconque du champ de caractère complet correspondant à l'extérieur de l'écran unitaire (ou à l'extérieur du champ actif (voir 5.3.3.6.2, CHAMP) si le champ de caractère se situe entièrement à l'intérieur du champ actif juste avant le déplacement), une séquence automatique APR (retour de chariot) et APD (interligne) est exécutée immédiatement. Lorsqu'une séquence explicite APR APD (ou APD APR) est reçue après l'exécution d'une séquence automatique APR APD, mais avant que l'origine du champ de caractère soit déplacée, alignée ou déterminée par toute autre commande ou séquence reçue, la séquence explicite APR APD (ou APD APR) doit être traitée comme une opération vide.

**5.3.2.3.7 Attributs de déplacement.** Les bits b4 et b3 du multiplet 2 permettent de définir la relation entre le mouvement du curseur et le mouvement de la position de dessin graphique, de la façon indiquée au tableau 10.

**Tableau 10**  
**Attributs de déplacement**

b4	b3	Attribut
0	0	Ensemble (par défaut)
0	1	Curseur en avance
1	0	Position de dessin en avance
1	1	Indépendamment

Lorsque le curseur et la position de dessin doivent se déplacer ensemble (00), la position de dessin suit le curseur chaque fois que celui-ci se déplace (par exemple, lorsque des caractères sont affichés), maintenant ainsi son alignement par rapport au curseur. Réciproquement, le curseur accompagne la position de dessin chaque fois que celle-ci se déplace (par exemple, dans le cas d'une primitive de dessin géométrique), maintenant également son alignement par rapport à la position de dessin.

Lorsque le curseur doit être en avance (01), la position de dessin suit le curseur, mais le curseur ne suit pas la position.

Lorsque la position de dessin doit être en avance (10), le curseur suit la position de dessin, mais la position ne suit pas le curseur.

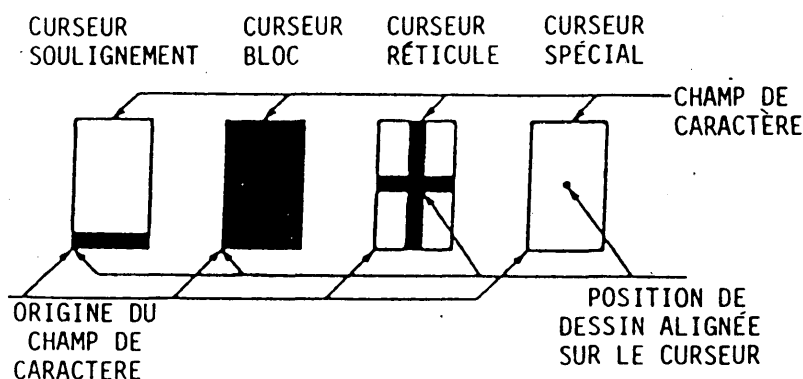
Lorsque la position de dessin et le curseur doivent se déplacer indépendamment (11), le déplacement de l'un n'influe pas sur l'emplacement de l'autre.

Le déplacement de la position de dessin ne doit jamais entraîner le curseur à un emplacement tel qu'une partie quelconque du champ de caractère indiqué par le curseur se retrouve à l'extérieur de l'écran unitaire. Dans une telle éventualité, le curseur doit être amené aussi près que possible de la position de dessin, sans toutefois transgresser la condition ci-dessus. Toute opération subséquente de positionnement relatif du curseur doit se faire par rapport à la position corrigée du curseur.

Le point d'alignement de la position de dessin correspond à l'origine du champ de caractère pour le curseur soulignement et le curseur bloc, et au centre du champ de caractère pour le curseur réticule et le curseur spécial. (Voir la figure 18.)

L'exécution de la commande TEXTE doit entraîner l'alignement de la position de dessin lorsque l'attribut de mouvement "ensemble" ou "curseur en avance" est opérant après l'exécution. L'exécution d'une commande TEXTE doit rester sans effet sur l'emplacement de l'origine du champ de caractère, sauf si l'exécution de la commande entraînerait une partie quelconque du champ de caractère à l'extérieur de l'écran unitaire. Dans ce cas, le curseur doit être positionné de la façon décrite ci-dessus.

**5.3.2.3.8 Styles de curseur.** Les bits b6 et b5 du multiplét 2 déterminent le style d'affichage du curseur, de la façon indiquée au tableau 11 et sur la figure 18.



Note: Les rectangles entourant les curseurs sont tracés aux fins d'illustration; ils ne font pas partie intégrante des styles de curseur.

Figure 18  
Styles de curseur



Tableau 11  
Styles de curseur

b6	b5	Style
0	0	Soulignement (par défaut)
0	1	Bloc
1	0	Réticule
1	1	Spécial

Le curseur indique l'emplacement où le caractère suivant doit être affiché. Le symbole du curseur soulignement est un trait simple tiré dans le bas du champ de caractère actuel, sur toute sa largeur. Le symbole du curseur bloc est un bloc plein dont les dimensions sont celles du champ de caractère actuel. Le symbole du curseur réticule est constitué d'une ligne verticale et d'une ligne horizontale qui se coupent 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. L'épaisseur du curseur soulignement et du curseur réticule, ainsi que la définition de la forme du curseur spécial, sont liées à la matérialisation.

**5.3.2.3.9 Dimensions du champ de caractère.** L'opérande à valeurs multiples suivant les deux premiers opérandes à format fixe donne la largeur (dx) et la hauteur (dy) du champ de caractère.

Si la valeur dx est négative, le profil des caractères se réfléchit de part et d'autre de l'axe vertical du champ de caractère. Si la valeur dy est négative, le profil des caractères se réfléchit de part et d'autre de l'axe horizontal du champ de caractère.

Si les dimensions du champ de caractère sont omises de l'opérande, les dimensions du champ de caractère actuel restent inchangées.

Les dimensions par défaut du champ de caractère sont  $dx = 1/40$  et  $dy = 5/128$ , compte tenu de la résolution physique.

La fonte et la position des caractères de texte alphanumériques à l'intérieur du champ de caractère sont liées à la matérialisation. Chacun de ces caractères doit entrer complètement dans l'aire définie par le champ de caractère actuel.

Les multiplats de données numériques suivant l'opérande à valeurs multiples seront normalisés plus tard; ils doivent rester sans effet.

#### 5.3.2.4 TEXTURE

**5.3.2.4.1 Format de la commande.** Cette commande permet de déterminer les attributs de texture qui sont appliqués au dessin subséquent des lignes, au rehaussement des aires remplies et aux motifs utilisés pour remplir les aires. Le code d'opération TEXTURE occupe un opérande à format fixe d'un multiplet, et il est suivi d'un opérande à valeurs multiples dont les interprétations sont données ci-dessous. (Voir la figure 19.)

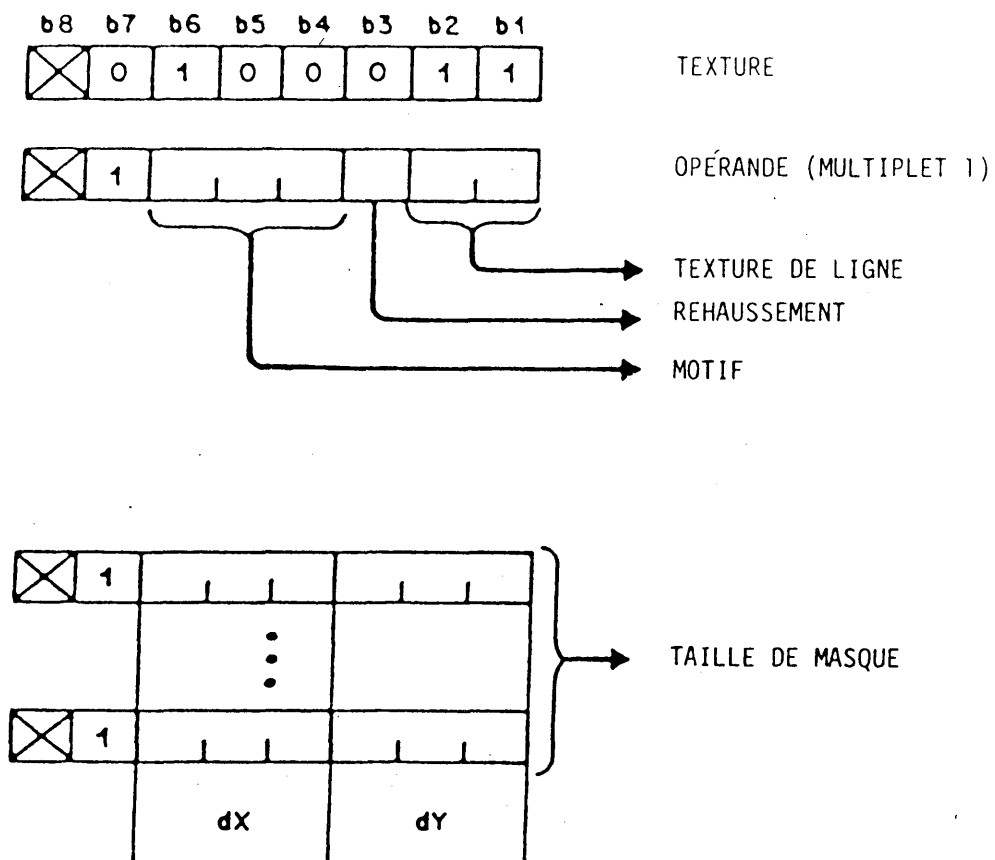


Figure 19  
Texture

**5.3.2.4.2 Texture de ligne.** Les bits b2 et b1 du multiplet 1 permettent d'établir l'attribut de texture de ligne, ce qui détermine le style des lignes et périmètres (mais pas des rehaussements) tracés à l'aide des `DOI LIGNE`, `ARC`, `RECTANGLE`, `POLYgone` et `INCRÉMENTATION LIGNE` (voir figure 20 et tableau 12). Les points ont les mêmes dimensions que le pel logique. L'espacement des points est égal à la largeur du pel logique pour les lignes horizontales, et il est égal à la hauteur du pel logique pour les lignes verticales. Pour les lignes horizontales, la hauteur des tirets est égale à la hauteur du pel logique, tandis que la largeur (longueur) et l'espacement des tirets sont égaux à trois fois la largeur du pel logique. Pour les lignes verticales, la largeur des tirets est égale à la largeur du pel logique, tandis que la hauteur (longueur) et l'espacement des tirets sont égaux à trois fois la hauteur du pel logique. L'espacement des points et tirets (ligne mixte) équivaut à l'espacement des points. En mode de couleur 2, la couleur de fond est utilisée entre les points et les tirets.

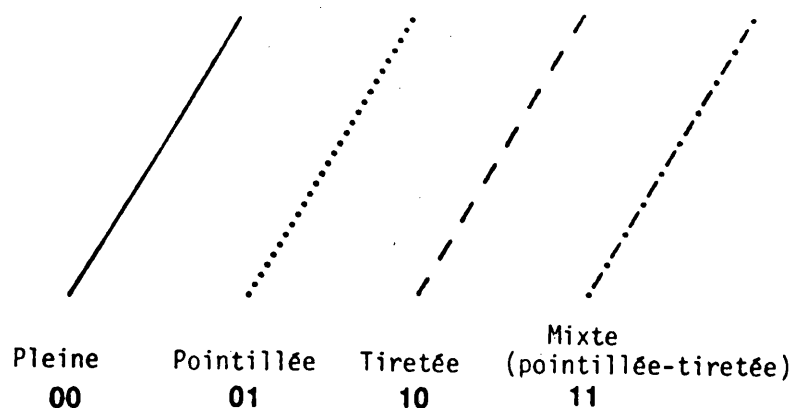


Figure 20

Texture de ligne

L'algorithme générateur des textures de ligne est lié à la matérialisation et devrait produire pour des lignes arbitraires des caractéristiques donnant un effet visuellement uniforme par rapport aux lignes horizontales et verticales, bien qu'une régularité parfaite ne soit pas garantie. Toutes les extrémités des lignes et arcs et tous les sommets des lignes d'incrémentations (drapeau de dessin au niveau vrai), des polygones d'incrémentations rehaussés, ainsi que des rectangles et polygones à périmètre ordinaire ou rehaussé doivent être tracés sans tenir compte de la texture de ligne utilisée.

**Note:** Lorsque la dimension  $dx$  du pel logique = 0, toutes les lignes non verticales sont pleines. Lorsque la dimension  $dy$  du pel logique = 0, toutes les lignes non horizontales sont pleines.

Tableau 12  
Texture de ligne

b2	b1	Texture
0	0	Pleine (par défaut)
0	1	Pointillée
1	0	Tirée
1	1	Mixte (pointillée-tirée)

**5.3.2.4.3 Rehaussement.** Le bit b3 du multiplet 1 détermine l'attribut de rehaussement. Lorsque le bit b3 est égal à 1, tous les rectangles, arcs, polygones et polygones d'incrémentations remplis sont dessinés en mode de rehaussement. Dans ce mode, les lignes ou arcs constituant les périmètres sont tracés avec la texture de ligne pleine (indépendamment de la texture actuelle), et les couleurs utilisées pour le pel logique actuel sont le noir nominal dans les modes de couleur 0 et 1 et la couleur de fond en mode de couleur 2. Le périmètre est la région tracée par le pel logique lorsqu'un arc, un rectangle, un polygone ou un polygone d'incrémentations est tracé. L'état par défaut de cet attribut est sans rehaussement (b3 = 0). (Voir la description des trois modes de couleur en 5.3.2.6.)

**5.3.2.4.4 Motif.** Les bits b6, b5 et b4 permettent de sélectionner le motif à utiliser pour remplir les rectangles, arcs, polygones et polygones d'incrémentations, de la façon indiquée au tableau 13 et sur la figure 21.



Figure 21

Motifs

Tableau 13  
Motifs

b6	b5	b4	Motif
0	0	0	Rempli (par défaut)
0	0	1	Hachures verticales
0	1	0	Hachures horizontales
0	1	1	Hachures verticales et horizontales croisées
1	0	0	Masque A
1	0	1	Masque B
1	1	0	Masque C
1	1	1	Masque D

La largeur et l'espacement des hachures verticales sont égaux à la largeur du pel logique. La hauteur et l'espacement des hachures horizontales sont égaux à la hauteur du pel logique. Le cadrage des motifs doit être maintenu sur toutes les figures lorsque la taille du pel logique est identique. Dans le cas des motifs prédéfinis, le motif rempli est toujours utilisé lorsque la taille du pel logique est (0,0). En mode de couleur 2, les aires de remplissage qui ne sont pas de la couleur de dessin sont de la couleur de fond.

Les masques de texture programmables A, B, C et D sont définis par la commande DÉF TEXTURE. Le motif par défaut des quatre masques de texture programmables est le motif vide, qui ne produit aucun remplissage dans les modes de couleur 0 et 1 et un remplissage avec la couleur de fond en mode de couleur 2.

**5.3.2.4.5 Taille de masque.** La paire de coordonnées suivant le premier multiplet de l'opérande spécifie la taille du masque (dx, dy) à utiliser pendant le processus de progression et répétition pour les masques A, B, C et D. Au cours de ce processus, le masque de texture sélectionné est mis à l'échelle de la taille spécifiée du masque, l'objet déterminé est recouvert logiquement de copies contiguës du masque, et tous les pixels visés par le motif du masque passent aux couleurs effectives. Le point de référence est l'origine (0,0) de l'écran unitaire de sorte que le cadrage des motifs est maintenu sur toutes les figures pour une taille de masque donnée.

La taille de masque par défaut est  $dx = 1/40$  et  $dy = 5/128$  (dimensions du champ de caractère par défaut), compte tenu de la résolution physique. Les bits de signe de dx et dy servent à réfléchir le motif à l'intérieur du champ de masque, comme dans le cas d'un champ de caractère de texte.

Si l'opérande de taille de masque n'est pas présent dans l'IDI TEXTURE, la taille de masque actuelle reste inchangée. Les multiplets de données numériques suivant l'opérande de taille de masque seront normalisés plus tard; ils doivent rester sans effet.

### 5.3.2.5 DÉTERMINATION COULEUR

5.3.2.5.1 La commande DÉTERMINATION COULEUR sert à spécifier les valeurs de couleur appliquées à toutes les commandes de dessin subséquentes et à tous les caractères des jeux primaire, supplémentaire, JCDR et mosaïque. Cette commande influe aussi sur les couleurs affichées antérieurement. Trois modes de couleur peuvent être sélectionnés; c'est de cette sélection que dépend l'interprétation précise des deux codes d'opération de commande de couleur, DÉTERMINATION COULEUR et SÉLECTION COULEUR. Le mode de couleur est établi à 0, 1 ou 2 par l'IDI SÉLECTION COULEUR de la façon décrite en 5.3.2.6. Le mode de couleur 0 est prévu pour les cas où la couleur de dessin est spécifiée directement sous forme de valeur de couleur. Dans ce mode, les couleurs sont définies implicitement dans la carte des couleurs. Les modes de couleur 1 et 2 permettent l'utilisation explicite d'une carte de couleurs, c'est-à-dire que la couleur de dessin est spécifiée par un nombre ordinal servant d'adresse dans un tableau de recherche qui contient la valeur de couleur actuelle.

Afin d'illustrer la différence entre les trois modes de couleur, considérons l'exemple d'un texte écrit. En mode de couleur 0, la couleur de dessin est déterminée directement, puis appliquée uniquement aux pixels d'avant-plan, soit aux seuls pixels qui constituent la configuration des caractères. En mode de couleur 1, la couleur est sélectionnée dans la carte des couleurs, puis de nouveau appliquée seulement aux pixels d'avant-plan. En mode de couleur 2, les couleurs de dessin et de fond sont toutes deux sélectionnées dans la carte des couleurs, puis appliquées respectivement aux pixels d'avant-plan et de fond.

La carte des couleurs permet de convertir, au moment de l'affichage, l'adresse de la carte des couleurs mémorisée pour chacun des pixels dans la plage d'affichage physique afin de donner la vraie valeur de couleur correspondant au pixel en question. Le nombre des bits ( $N$ ) de l'adresse de la carte des couleurs (c'est-à-dire le nombre de bits par pixel dans la mémoire d'affichage) est, par conception, plus faible que le nombre de bits ( $M$ ) de la valeur de couleur réelle mémorisée dans la carte des couleurs (c'est-à-dire la largeur de la carte des couleurs). Il s'ensuit notamment une augmentation du nombre total des couleurs d'affichage possibles (jusqu'à  $2^M$ ) sans accroissement de la capacité de mémoire de l'affichage, mais avec la restriction qu'au plus  $2^N$  couleurs peuvent être affichées simultanément.

La définition complète d'une couleur dans les modes de couleur 1 et 2 nécessite deux étapes: la spécification des valeurs de couleur mémorisées dans la carte des couleurs et la spécification de l'adresse de la carte des couleurs (soit un nombre ordinal) à associer à la couleur de dessin. Dans les modes de couleur 1 et 2, les commandes DÉTERMINATION COULEUR et SÉLECTION COULEUR remplissent respectivement la première fonction et la seconde fonction. À noter que la carte des couleurs vise tout l'affichage. Une modification de la carte des couleurs se reflète immédiatement sur la couleur de tous les pixels associés à l'adresse de la carte des couleurs pointant sur l'entrée de la carte qui a été changée.

Le mode de couleur 0 fait également appel à la carte des couleurs. Si la couleur spécifiée par la commande DÉTERMINATION COULEUR du mode de couleur 0 a déjà été spécifiée dans la carte des couleurs, l'adresse de la couleur de dessin devient l'adresse la plus basse de la couleur en question et la carte des couleurs reste inchangée. Si cette couleur n'a pas déjà été spécifiée dans la carte des couleurs, le mode de couleur 0 fait appel à l'adresse la plus basse qui est restée inutilisée (par l'intermédiaire d'une commande DÉTERMINATION COULEUR du mode de couleur 0, 1 ou 2, ou d'une commande SÉLECTION COULEUR du mode de couleur 1 ou 2) depuis la dernière commande INITIALISATION rétablissant la carte des couleurs par défaut; l'adresse utilisée doit toutefois être une adresse autre que celles du noir nominal ou du blanc nominal. Si aucune adresse n'est disponible, la carte des couleurs doit rester inchangée et la couleur de dessin doit être établie selon une méthode liée à la matérialisation.

Il est recommandé d'appliquer la relation ci-dessous entre le nombre de bits (N) de l'adresse de la carte des couleurs et le nombre de bits (M) des valeurs de couleur mémorisées dans la carte des couleurs.

$$M \geq 3 (N-1)$$

Le code d'opération DÉTERMINATION COULEUR, qui occupe un opérande à valeurs multiples, est illustré sur les figures 22 et 23. L'opérande de valeur de couleur sert à définir une couleur de la façon indiquée sur la figure 23.

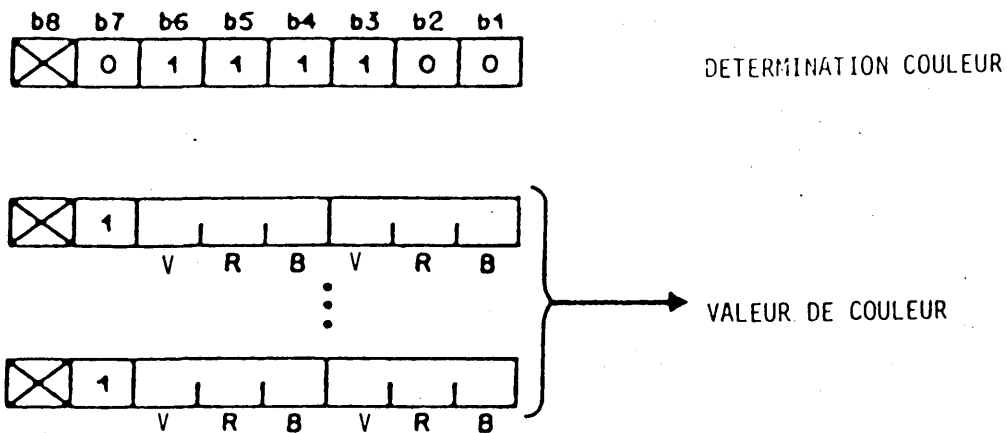


Figure 22  
Détermination couleur

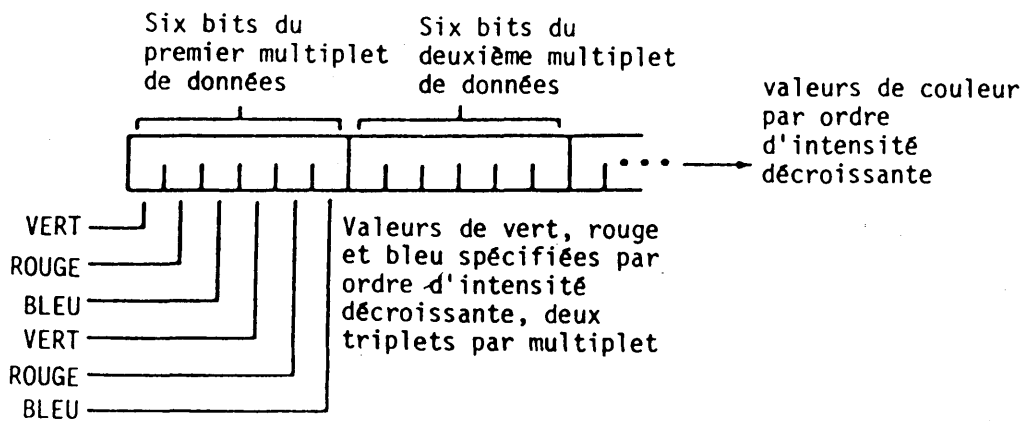


Figure 23  
Structure de codage des couleurs



En mode de couleur 0, la couleur de dessin se trouve ainsi établie. Cette couleur est appliquée à tous les dessins de caractères alphanumériques et d'images reçus subséquemment, jusqu'à ce qu'elle soit changée par une autre commande DÉTERMINATION COULEUR, la commande INITIALISATION décrite en 5.3.2.9, ou le caractère de commande NSR décrit en 6.1.6.5. La couleur de dessin par défaut est le blanc en mode de couleur 0. Une couleur de fond ne peut pas être spécifiée en mode de couleur 0, c'est-à-dire que les configurations de caractère et dessins d'image se substituent tout simplement au contenu existant de la plage d'affichage physique, qui reste autrement inchangé.

Dans les modes de couleur 1 et 2, la commande DÉTERMINATION COULEUR sert à charger les valeurs de couleur dans la carte des couleurs. L'adresse utilisée pour l'entrée à charger est celle indiquée par la couleur de dessin (qui doit avoir été établie précédemment à l'aide de la commande SÉLECTION COULEUR).

Si la longueur maximale (nombre de bits) de l'entrée que peut recevoir la carte des couleurs est plus faible que le nombre des bits fournis par l'opérande DÉTERMINATION COULEUR, l'opérande est tronqué et seuls les bits de poids fort sont utilisés. Si la longueur maximale de l'entrée que peut recevoir la carte des couleurs est plus grande que le nombre des bits fournis par l'opérande DÉTERMINATION COULEUR, des zéros non significatifs sont ajoutés au cours du processus de présentation à la réception. Pour chaque primaire, la fraction maximale de couleur pouvant être atteinte compte tenu du nombre de bits spécifiés dans l'opérande de valeur de couleur doit être interprétée comme l'intensité maximale et les valeurs intermédiaires doivent être distribuées également entre zéro et l'intensité maximale.

Si la commande DÉTERMINATION COULEUR est répétée implicitement par l'émission de données numériques supplémentaires, l'adresse de l'entrée de couleur à changer est incrémentée automatiquement avant l'exécution du nouveau code d'opération. L'algorithme d'incrémentement a pour effet de remplacer le zéro de plus fort poids par un 1 et de remplacer par un 0 tous les 1 situés plus à gauche. Par exemple, l'adresse 010100 de la carte des couleurs serait incrémentée à 110100, puis à 001100. Cette double incrémentation n'influe pas sur l'adresse de la carte des couleurs associée à la couleur de dessin. Le processus d'incrémentement s'arrête et les données d'opérande subséquentes restent sans effet une fois atteinte la limite physique (une seulement) de la carte des couleurs matérialisée.

**Note:** Cet algorithme d'incrémentement permet d'obtenir la même présentation sur un appareil comportant un nombre d'entrées de carte des couleurs plus élevé que ne l'avait supposé le fournisseur. En outre, la disposition attentive des couleurs semblables pour des entrées adjacentes de la carte des couleurs permet d'obtenir une présentation raisonnable sur un appareil comportant moins d'entrées de carte des couleurs que ne l'avait supposé le fournisseur.

L'absence d'opérande à la suite du code d'opération DÉTERMINATION COULEUR établit la couleur transparente. Lorsque la couleur transparente est utilisée, tout plan de rang inférieur apparaît sur l'affichage. (Ces plans de rang inférieur pourraient correspondre à des plans de valeur Z plus basse dans une architecture de terminal à plans multiples, ou à un signal vidéo analogique dans des applications où l'affichage vidéotex est superposé à une image de télévision normale, par exemple pour le sous-titrage.) S'il n'existe aucun plan de rang inférieur ou si la transparence n'est pas matérialisée, la couleur transparente est interprétée comme le noir.

5.3.2.5.2 Le contenu par défaut de la carte des couleurs est défini selon l'algorithme décrit ci-dessous:

N = nombre de bits d'adresse de la carte des couleurs  
 $2^N$  = capacité de la carte des couleurs  
M = nombre de bits de valeur de couleur (c'est-à-dire largeur de la carte des couleurs)  
 $M \geq 3(N-1)$  selon les spécifications précédentes

La première moitié de la carte des couleurs par défaut sert à mémoriser une échelle des gris complète à espacements uniformes. Il s'agit d'un jeu ordonné de couleurs où  $V = R = B$ .

(À noter que si  $M = 3(N-1)$ , il devrait y avoir exactement  $(2^N)/2$  niveaux de gris, y compris le blanc et le noir.) La seconde moitié de la carte des couleurs par défaut sert à mémoriser une gamme complète de nuances espacées également par rapport au périmètre du cercle des nuances. Le cercle des nuances illustré sur la figure 24 est défini à partir des trois couleurs primaires (vert, rouge et bleu) équidistantes sur la circonférence d'un cercle dont le bleu se trouve à 0 degré, le rouge à 120 degrés, et le vert à 240 degrés. Toutes les autres nuances peuvent être obtenues par diverses combinaisons de ces trois primaires dans des proportions qui dépendent de la position de la nuance désirée sur le cercle. L'algorithme permettant d'obtenir les valeurs VRB des nuances par défaut, espacées également sur la circonférence du cercle à partir de 0 degré et dans le sens antihoraire, est le suivant:

Soit:

h = nuance désirée  
ang h = angle de h  
 $P_1$  = primaire la plus rapprochée de h  
ang  $P_1$  = angle de  $P_1$   
 $P_2$  = primaire intermédiaire par rapport à h  
ang  $P_2$  = angle de  $P_2$   
 $P_3$  = primaire la plus éloignée de h

Les valeurs des primaires du système VRB qui doivent être combinées pour donner la nuance h sont les suivantes:

- 1)  $P_1 = 1$  (tous les bits remis à 1)
- 2)  $P_2 = \frac{|\text{ang } h - \text{ang } P_1|}{60 \text{ degrés}}$
- 3)  $P_3 = 0$  (tous les bits remis à 0)

La valeur de  $P_2$  est ensuite normalisée, c'est-à-dire multipliée par la valeur de couleur maximale pouvant être mémorisée pour la primaire en question. Par exemple, si trois bits sont disponibles pour mémoriser la primaire, le résultat fourni par l'équation ci-dessus de  $P_2$  est multiplié par  $7/8$  (fraction binaire maximale exprimable avec trois bits), puis arrondi à trois positions.

À titre d'exemple, la carte des couleurs par défaut pour  $N = 4$  et  $M = 9$  est illustrée sur la Figure 25.

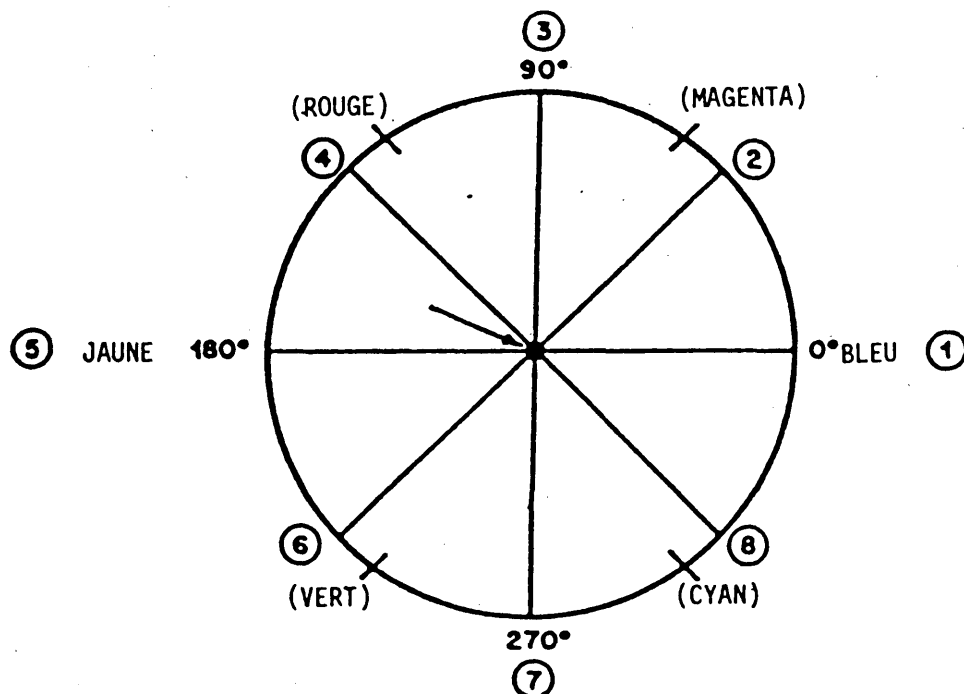


Figure 24

Sélection des couleurs par défaut

ADRESSE DE LA CARTE DES COULEURS	VALEURS DE COULEUR		
	V	R	B
0 0 0 0	0 0 0	0 0 0	0 0 0
0 0 0 1	0 0 1	0 0 1	0 0 1
0 0 1 0	0 1 0	0 1 0	0 1 0
0 0 1 1	0 1 1	0 1 1	0 1 1
0 1 0 0	1 0 0	1 0 0	1 0 0
0 1 0 1	1 0 1	1 0 1	1 0 1
0 1 1 0	1 1 0	1 1 0	1 1 0
0 1 1 1	1 1 1	1 1 1	1 1 1
1 0 0 0	0 0 0	0 0 0	1 1 1
1 0 0 1	0 0 0	1 0 1	1 1 1
1 0 1 0	0 0 0	1 1 1	1 0 0
1 0 1 1	0 1 0	1 1 1	0 0 0
1 1 0 0	1 1 1	1 1 1	0 0 0
1 1 0 1	1 1 1	0 1 0	0 0 0
1 1 1 0	1 1 1	0 0 0	1 0 0
1 1 1 1	1 0 1	0 0 0	1 1 1

NOIR  
NOMINAL

↑

ÉCHELLE  
DES  
GRIS

↓

BLANC  
NOMINAL

↑

NUANCES

↓

Figure 25

Carte des couleurs par défaut pour N = 4 et M = 9

### 5.3.2.6 SÉLECTION COULEUR

5.3.2.6.1 Le code d'opération SÉLECTION COULEUR sert à établir le mode de couleur ainsi qu'à sélectionner la couleur de dessin pour les modes 1 et 2, et la couleur de fond pour le mode 2. (Voir la figure 26).

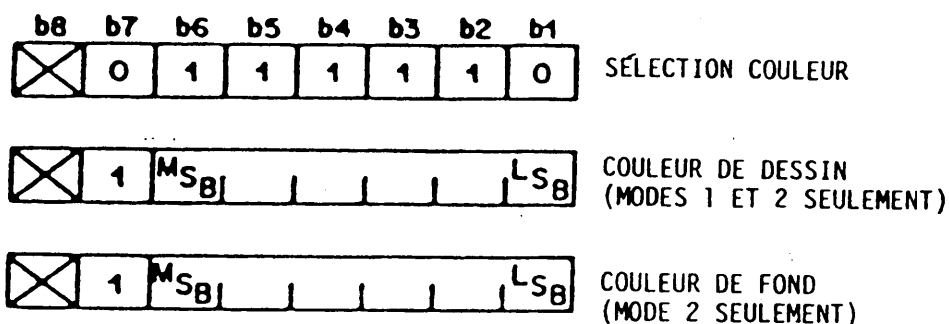


Figure 26  
Sélection couleur

Le code d'opération SÉLECTION COULEUR peut occuper zéro, un ou deux opérandes à valeur simple. Les multiplats de données numériques supplémentaires seront normalisés plus tard et doivent rester sans effet. Lorsque le code d'opération SÉLECTION COULEUR n'est suivi d'aucun opérande, le mode de couleur 0 est indiqué. Le terminal reste en mode de couleur 0 jusqu'à la réception d'une autre commande SÉLECTION COULEUR et de ses opérandes, ou jusqu'au changement du mode de couleur au moyen d'une commande INITIALISATION (décrite en 5.3.2.9). En mode de couleur 0, la commande DÉTERMINATION COULEUR sert à établir la couleur de dessin, de la façon décrite dans la section précédente, et la commande SÉLECTION COULEUR reste inutilisée. Aucune couleur de fond n'est spécifiée en mode de couleur 0, et les caractères alphanumériques ainsi que les dessins d'image se substituent tout simplement au contenu existant de l'écran d'affichage physique, seulement aux endroits où la couleur de dessin est appliquée.

Si le nombre matérialisé des bits d'adresse de la carte des couleurs (N) est inférieur au nombre des bits concaténés disponibles dans un opérande à valeur simple (6, 12, 18 ou 24, selon l'opérande de longueur à valeur simple de la commande DOMAINE reçue en dernier, ou la valeur par défaut de 6 lorsque aucune commande DOMAINE n'a été reçue depuis une commande INITIALISATION ou NSR), seuls les bits de poids fort sont significatifs. En d'autres termes, le nombre de bits nécessaire pour spécifier l'adresse de la carte des couleurs est justifié à gauche dans un opérande à valeur simple. Si l'on considère par exemple un opérande à valeur simple occupant par défaut un multiplat et une adresse de carte des couleurs de 4 bits ( $N = 4$ ), le processus de présentation à la réception répond alors aux bits b6 à b3 et ne tient pas compte des bits b2 et b1 dans chaque opérande d'adresse de la carte des couleurs associé aux commandes DÉTERMINATION COULEUR et CLIGNOTEMENT. Si le nombre matérialisé des bits d'adresse de la carte des couleurs (N) est supérieur au nombre des bits concaténés disponibles dans un opérande à valeur simple (6, 12, 18 ou 24), des zéros non significatifs sont ajoutés dans le processus de présentation à la réception.

5.3.2.6.2 Si le code d'opération SÉLECTION COULEUR est suivi d'un seul opérande, le mode de couleur 1 est indiqué (sans incidence sur la carte des couleurs). Le terminal reste en mode de couleur 1 jusqu'à la réception d'une autre commande SÉLECTION COULEUR à 0 ou 2 opérandes, ou jusqu'au changement du mode de couleur au moyen d'une commande INITIALISATION ou NSR. En mode de couleur 1, l'opérande unique suivant le code d'opération SÉLECTION COULEUR sert à établir la couleur de dessin appliquée à l'information de texte alphanumérique et d'image reçue subséquent. Il convient à nouveau de noter que la couleur de dessin dans ce cas est un nombre ordinal représentant l'adresse de la carte des couleurs à laquelle a été précédemment ou sera ultérieurement chargée la valeur de couleur réelle au moyen d'une commande DÉTERMINATION COULEUR. Aucune couleur de fond n'est spécifiée en mode de couleur 1, et les caractères alphanumériques ainsi que les dessins d'image se substituent tout simplement au contenu existant de l'écran d'affichage physique, seulement aux endroits où la couleur de dessin est appliquée.

5.3.2.6.3 Si le code d'opération SÉLECTION COULEUR est suivi de deux opérandes, le mode de couleur 2 est indiqué. À nouveau, le terminal reste en mode de couleur 2 jusqu'à la réception d'une autre commande SÉLECTION COULEUR à 0 ou 1 opérande, ou jusqu'au changement du mode de couleur au moyen d'une commande INITIALISATION ou NSR. En mode de couleur 2, le premier opérande suivant le code d'opération SÉLECTION COULEUR sert à établir la couleur de dessin et le deuxième, la couleur de fond. Les caractères reçus en mode de couleur 2 sont tracés par l'application de la couleur de dessin sur la couleur de fond, qui occupe le reste du champ de caractère. La couleur de fond ne touche pas la partie de l'espace entre les caractères qui n'est pas incluse dans le champ de caractère. Dans le cas spécial où les deux opérandes sont identiques, c'est-à-dire lorsque la couleur de dessin spécifiée est la même que la couleur de fond, la couleur de dessin reste plutôt à sa valeur actuelle et seule la couleur de fond passe à la valeur spécifiée. La couleur de fond est également utilisée comme couleur de rehaussement et comme couleur alternante dans les motifs des lignes et aires, comme l'indique la description donnée en 5.3.2.4.

### 5.3.2.7 CLIGNOTEMENT

5.3.2.7.1 La commande CLIGNOTEMENT est utilisée pour qu'une entrée de la carte des couleurs alterne périodiquement entre deux couleurs.

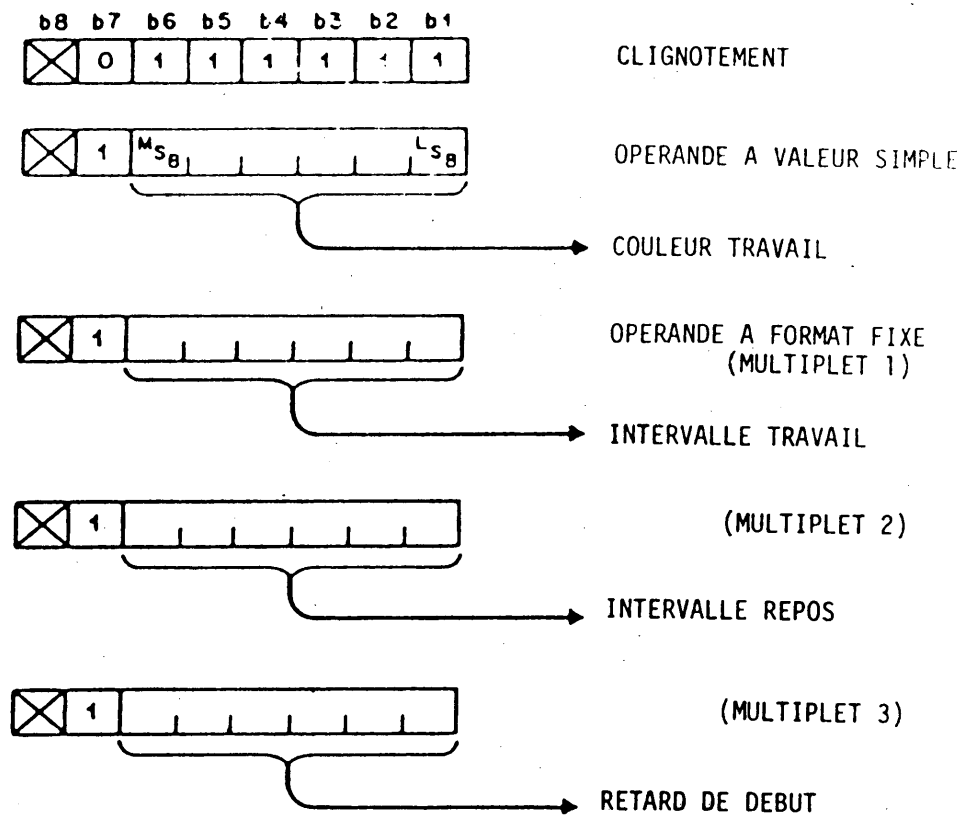


Figure 27  
Clignotement

5.3.2.7.2 Cette alternance est réalisée par un processus de clignotement: le contenu actuel d'une entrée de la carte des couleurs, appelée couleur travail, est substitué périodiquement à la couleur de dessin effective actuelle, appelée couleur repos. La couleur travail est présente pendant une période appelée intervalle travail et la couleur repos, pendant une période appelée intervalle repos. Les intervalles travail et repos alternent en cadence, en commençant par l'intervalle travail. Un retard de début peut également être spécifié; il s'agit alors de retarder le début de l'intervalle travail par rapport au début de l'intervalle travail du dernier processus de clignotement actif défini. Une spécification de retard de début reste sans effet lorsque aucun processus de clignotement actif n'est en cours. Lorsque des processus de clignotement multiples comportent des intervalles travail ou repos qui se terminent simultanément, ces processus sont traités séquentiellement, du processus défini en dernier au processus défini en premier. Dans ce cas, l'entrée utilisée à chaque processus de clignotement vient de la carte des couleurs obtenue à la suite du processus de clignotement exécuté précédemment.

5.3.2.7.3 Le premier opérande à valeur simple suivant le code d'opération CLIGNOTEMENT est la spécification de la couleur travail, donnée sous forme d'adresse de la carte des couleurs (voir 5.3.2.6.1). L'opérande suivant à format fixe est l'intervalle travail spécifié en dixièmes de seconde. Seuls les bits b6 à b1 sont utilisés pour cette spécification. De même, l'opérande suivant à format fixe spécifie l'intervalle repos. Le quatrième opérande à format fixe spécifie le retard de début, également en dixièmes de seconde. L'omission de ce multiplète indique un retard de début nul. S'il n'existe actuellement aucun processus de clignotement actif, le multiplète reste sans effet. Un intervalle travail ou repos nul est interprété comme la fin de tout processus de clignotement actif appliqué à la paire des couleurs travail/repos (voir figure 27).

5.3.2.7.4 La définition d'un processus de clignotement entre deux couleurs, une de travail et une de repos, met automatiquement fin à tout autre processus de clignotement défini précédemment et mettant en jeu la même paire de couleurs. L'absence d'opérande à la suite du code d'opération CLIGNOTEMENT met fin à tous les processus de clignotement utilisant la couleur de dessin actuelle comme couleur repos. La couleur repos originale doit être rétablie (à moins qu'elle ait été changée explicitement par une commande DÉTERMINATION COULEUR) une fois qu'ont pris fin tous les processus de clignotement mettant en jeu cette couleur repos.

5.3.7.2.5 Si des données supplémentaires suivent les spécifications complètes d'un processus de clignotement, la commande CLIGNOTEMENT est répétée implicitement, l'adresse de la couleur repos étant automatiquement incrémentée (comme en 5.3.2.5.1) avant l'exécution du nouveau code d'opération. La couleur de dessin n'est pas touchée par l'incrémementation.



### 5.3.2.8 ATTENTE

5.3.2.8.1 La commande ATTENTE permet de produire un retard de durée déterminée dans le traitement.

L'intervalle d'attente commence à la dernière des deux éventualités suivantes: la fin de l'exécution de la commande précédente ou la réception de la commande ATTENTE.

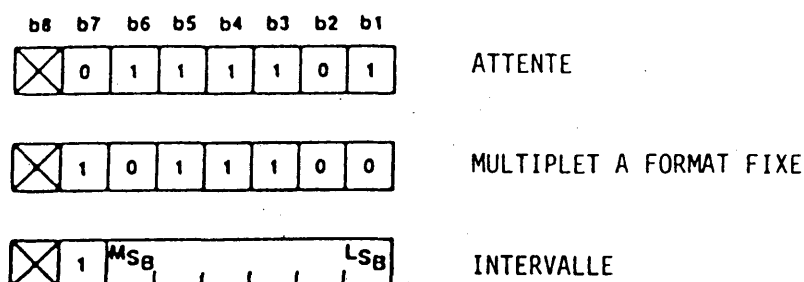


Figure 28  
Attente

5.3.2.8.2 Le premier multipliet des données d'opérande suivant le code d'opération ATTENTE doit être conforme au format donné sur la figure 28. Lorsqu'une autre combinaison de bits suit le code d'opération ATTENTE, toute la commande doit être traitée comme une opération vide, la normalisation étant remise à plus tard. Le multipliet suivant des données d'opérande donne le retard en dixièmes de seconde (64 valeurs codées binaires). Seuls les bits b1 à b6 sont utilisés à cette fin. Lorsque des multiplats supplémentaires sont donnés à la suite, ils sont traités comme des périodes additionnelles d'attente, chaque période étant spécifiée indépendamment par chacun des multiplats. Un opérande de 0 indique un intervalle d'attente lié à la matérialisation et compris entre 0 et 1/10 de seconde (inclusivement).

### 5.3.2.9 INITIALISATION

5.3.2.9.1 La commande INITIALISATION sert à réinitialiser sélectivement à leur valeur par défaut les paramètres de commande et d'attribut, à effacer l'écran, à déterminer la couleur de la plage périphérique, à ramener le curseur à sa position initiale, et à remettre à 0 le JCDR, les attributs de texture, les macro-codes et les champs sans protection (description en 6.2). Le code d'opération INITIALISATION occupe un opérande à format fixe de 2 multipléts. L'exécution des initialisations se fait dans l'ordre suivant: multipléte 1, du bit de poids faible (b1) au bit de poids fort (b6), puis multipléte 2, du bit de poids faible (b1) ou bit de poids fort (b6). Le code d'opération INITIALISATION et son opérande sont illustrés ci-dessous. (Voir la figure 29.)

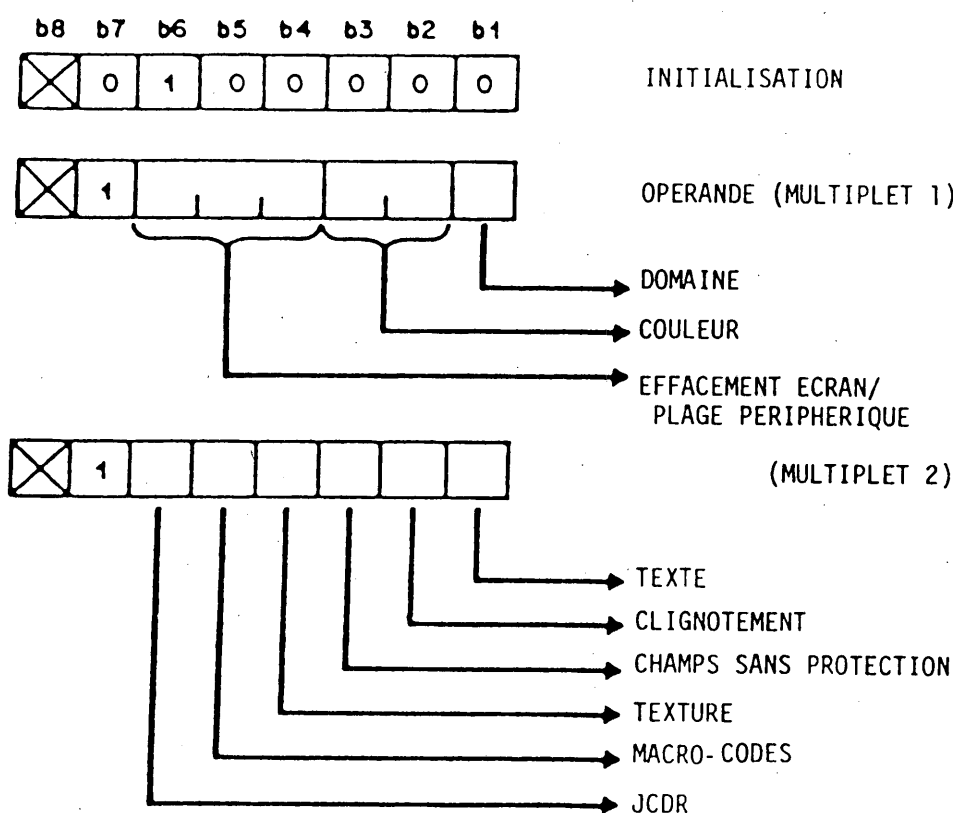


Figure 29  
Initialisation

**5.3.2.9.2** Multiplét 1 d'opérande de la commande INITIALISATION. Lorsque le bit b1 du multiplét 1 est à 1, les paramètres DOMAINE sont initialisés à leur valeur par défaut. Lorsque le bit b1 est à 0, les paramètres DOMAINE restent inchangés.

Les bits b3 et b2 du multiplét 1 changent le mode de couleur et la couleur de dessin actuelle de la façon indiquée au tableau 14.

**Tableau 14**

**Initialisation du mode de couleur**

b3	b2	Mode de couleur
0	0	Pas de changement
0	1	Sélection du mode de couleur 0, passage aux couleurs par défaut de la carte des couleurs et passage au blanc de la couleur de dessin effective.
1	0	Sélection du mode de couleur 1 et passage aux couleurs par défaut de la carte des couleurs. Si ces bits sont traités en mode de couleur 0, ils ont le même effet que "11".
1	1	Sélection du mode de couleur 1, passage aux couleurs par défaut de la carte des couleurs et passage au blanc de la couleur de dessin effective.

Les bits b6, b5 et b4 du multiplét 1 permettent d'effacer la plage d'affichage et la plage périphérique pour passer aux couleurs indiquées au tableau 15.

La plage périphérique entoure la plage d'affichage et ne peut prendre qu'une seule couleur à la fois.

**Tableau 15**

**Initialisation de la plage d'affichage et de la plage périphérique**

b6	b5	b4	Couleurs de la plage d'affichage et de la plage périphérique
0	0	0	Pas de changement.
0	0	1	Passage de la plage d'affichage au noir nominal.
0	1	0	Passage de la plage d'affichage à la couleur de dessin actuelle.
0	1	1	Passage de la plage périphérique au noir nominal.
1	0	0	Passage de la plage périphérique à la couleur de dessin actuelle.
1	0	1	Passage de la plage d'affichage et de la plage périphérique à la couleur de dessin actuelle.
1	1	0	Passage de la plage d'affichage à la couleur de dessin actuelle et de la plage périphérique au noir nominal.
1	1	1	Passage de la plage d'affichage et de la plage périphérique au noir nominal.

5.3.2.9.3 Multiplet 2 d'opérande de la commande INITIALISATION. Si le bit b1 du multiplet 2 est à 1, le curseur est ramené à sa position initiale (position du caractère supérieur gauche sur la plage d'affichage), et tous les paramètres de texte (code d'opération TEXTE, jeu C1 et champ actif) sont initialisés à leur valeur par défaut. Si b1 est à 0, les paramètres de texte et la position du curseur restent inchangés.

Si le bit b2 du multiplet 2 est à 1, tout clignotement prend fin. Si b2 est à 0, tout clignotement se poursuit.

Si le bit b3 du multiplet 2 est à 1, tous les champs sans protection passent à l'état de protection, mais le contenu affiché reste inchangé. Il y a toutefois perte des définitions de champ (sauf pour ce qui est du champ actif), ainsi que de toutes les structures de données maintenues pour fins d'édition par l'utilisateur et de transmission. Si le bit b3 est à 0, les champs sans protection restent dans le même état.

Si le bit b4 du multiplet 2 est à 1, tous les attributs de texture sont initialisés à leur valeur par défaut. Les quatre masques de texture programmables ne sont pas touchés. Si le bit b4 est à 0, les attributs de texture actuels restent inchangés.

Si le bit b5 du multiplet 2 est à 1, tous les macro-codes, y compris ceux de transmission, sont supprimés. Si b5 est à 0, les macro-codes ne sont pas touchés.

Si le bit b6 du multiplet 2 est à 1, tous les caractères du JCDR sont remis à 0, c'est-à-dire que toutes les positions de caractère prennent la valeur du caractère d'espacement. Si b6 est à 0, les caractères du JCDR ne sont pas touchés.

Si la commande INITIALISATION est reçue sans opérande, elle est interprétée en supposant que les bits b6 à b1 des deux multiplets sont à 0. Si seulement un multiplet est reçu, les bits b6 à b1 du second multiplet sont supposés à 0. Si plus de 2 multiplets de données sont reçus, les multiplets supplémentaires doivent rester sans effet, leur normalisation étant remise à plus tard.

Les macro-instructions de transmission et du JCDR sont respectivement données en 5.5 et 5.6.

### 5.3.3 Primitives de dessin géométriques

Il est à noter que les dessins de la présente section sont stylisés et n'ont pas pour but de montrer une image réelle (voir figure 15).

#### 5.3.3.1 POINT

5.3.3.1.1 La commande POINT permet d'exécuter deux opérations de dessin géométrique fondamentales, soit l'établissement des coordonnées auxquelles doit commencer un dessin et la production d'un point. Une paire de coordonnées accompagne cette commande pour déterminer la position de dessin. Facultativement, un point peut être tracé (c'est-à-dire rendu visible) à la position spécifiée par les coordonnées. Les coordonnées peuvent être spécifiées sous forme de position absolue (X, Y) ou de déplacement relatif (dx, dy) par rapport à la position de dessin actuelle. (Voir la figure 30.)

Une série de coordonnées suivant un code d'opération POINT peut servir à dresser un graphique point par point. La position de dessin finale, c'est-à-dire la position de dessin à la fin de l'exécution de la commande POINT, est la dernière position spécifiée.

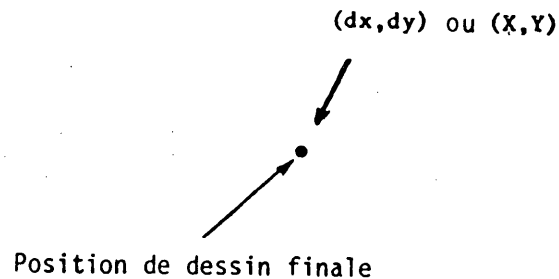
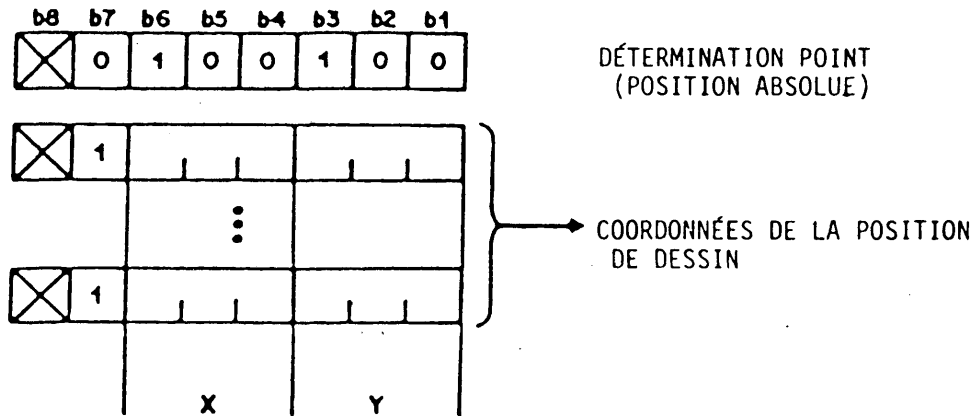


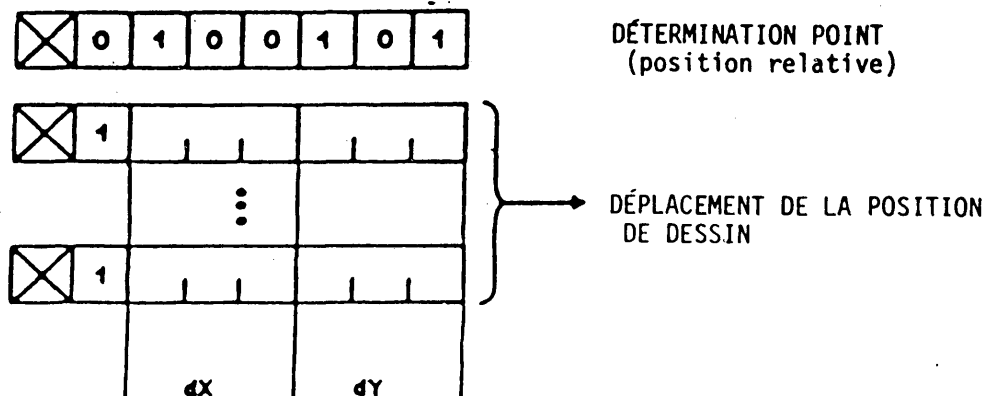
Figure 30  
POINT

**5.3.3.1.2 DÉTERMINATION POINT (position absolue, invisible).** Cette commande détermine la position de dessin selon les coordonnées absolues spécifiées. Le point n'est pas dessiné. (Voir la figure 31.)



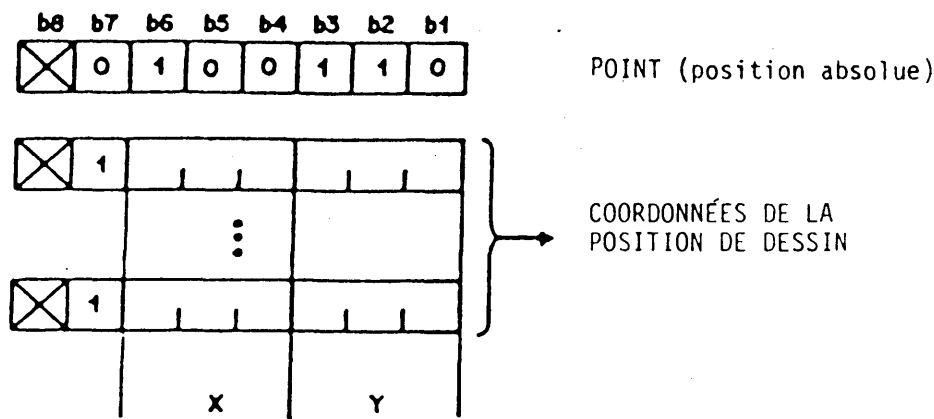
**Figure 31**  
DÉTERMINATION POINT (position absolue, invisible)

**5.3.3.1.3 DÉTERMINATION POINT (position relative, invisible).** Cette commande détermine la position de dessin selon les coordonnées obtenues en ajoutant le déplacement spécifié aux coordonnées de la position de dessin actuelle. Le point n'est pas dessiné. (Voir la figure 32.)



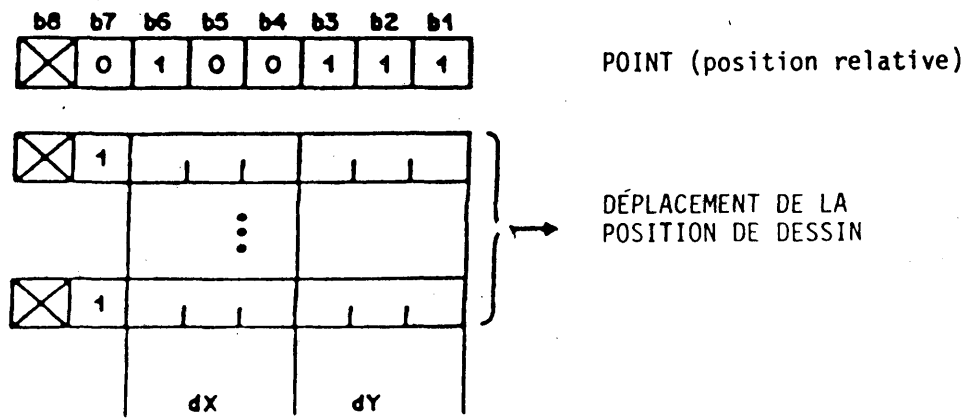
**Figure 32**  
DÉTERMINATION POINT (position relative, invisible)

**5.3.3.1.4 POINT (position absolue, visible).** Cette commande détermine la position de dessin selon les coordonnées absolues spécifiées et dessine un point dont la grosseur est déterminée par la taille du pel logique et dont la couleur est déterminée par la couleur de dessin. (Voir la description du pel logique en 5.3.2.2.) (Voir la figure 33.)



**Figure 33**  
POINT (position absolue, visible)

**5.3.3.1.5 POINT (position relative, visible).** Cette commande détermine la position de dessin selon les coordonnées obtenues en ajoutant le déplacement spécifié aux coordonnées de la position de dessin actuelle, et dessine un point dont la grosseur est déterminée par la taille du pel logique et dont la couleur est déterminée par la couleur de dessin. (Voir la figure 34.)



**Figure 34**  
POINT (position relative, visible)

### 5.3.3.2 LIGNE

5.3.3.2.1 La commande LIGNE constitue une opération fondamentale de dessin géométrique. La direction et la longueur d'une ligne sont spécifiées par ses positions de début et de fin. La position de début est spécifiée explicitement par la commande LIGNE ou en tant que position de dessin actuelle. La position de fin est spécifiée par un déplacement relatif (dx, dy) par rapport à la position de début, ou par des coordonnées absolues (X, Y). À la fin du tracé d'une ligne, la position de dessin coïncide avec la position de fin. La couleur utilisée pour tracer la ligne entre la position de début et la position de fin est la couleur actuelle; la largeur de la ligne est déterminée par la taille du pel logique, et sa texture par l'attribut de texture actuel. (Voir la figure 35.)

Tout comme le code d'opération POINT, la commande LIGNE peut servir à tracer un graphique à partir d'un tableau de nombres décrivant des coordonnées absolues ou relatives.

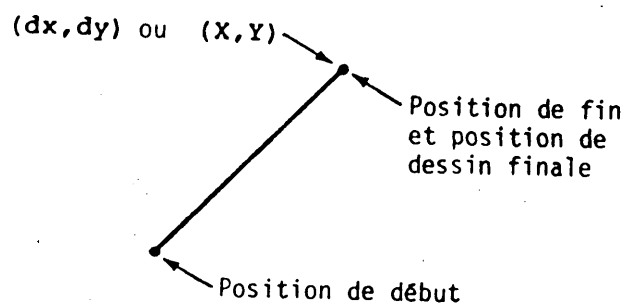
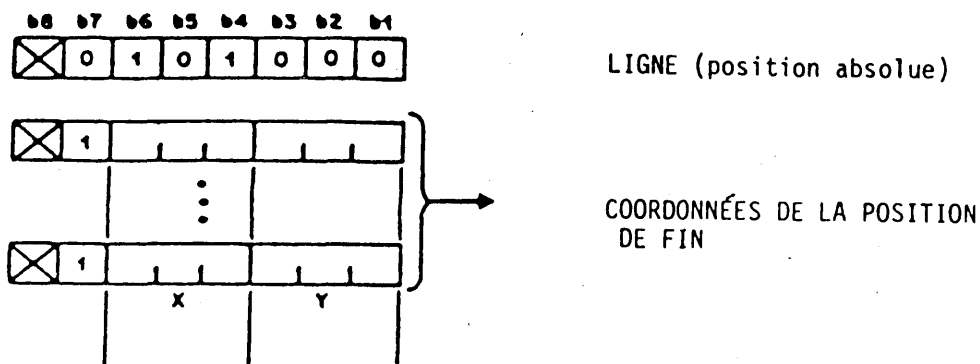


Figure 35

LIGNE

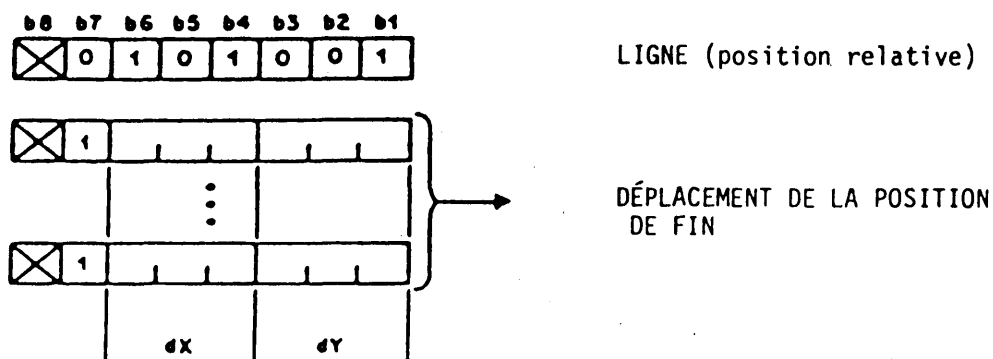


**5.3.3.2.2 LIGNE (position absolue).** La position de début est la position de dessin actuelle. La position de fin est spécifiée par des coordonnées absolues. (Voir la figure 36.)



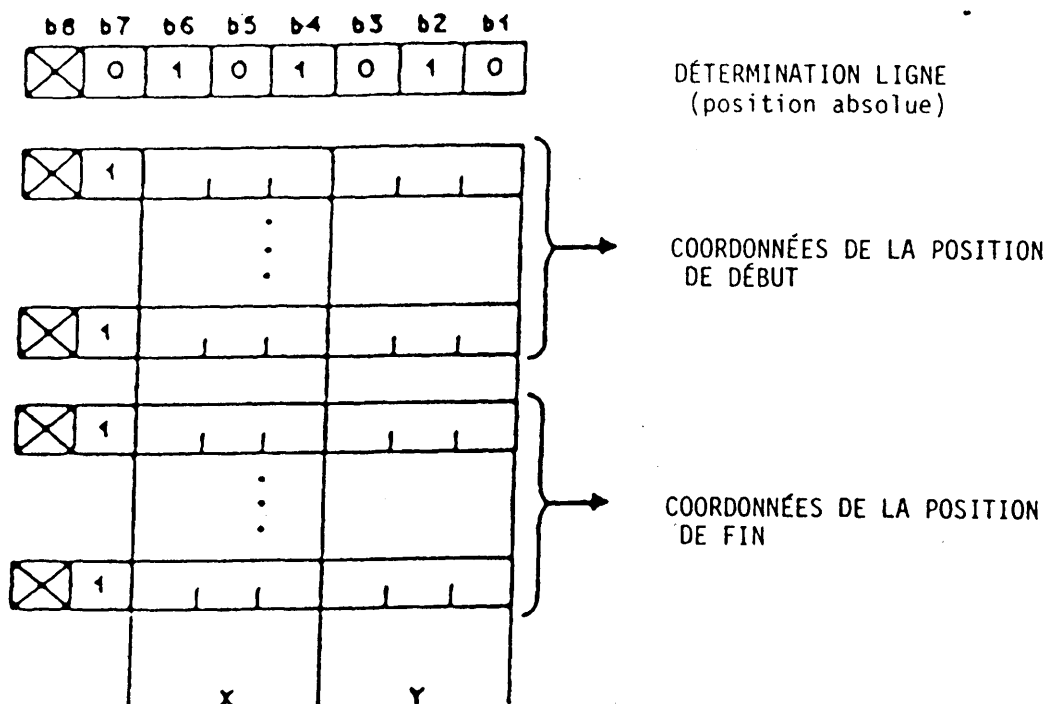
**Figure 36**  
LIGNE (position absolue)

**5.3.3.2.3 LIGNE (position relative).** La position de début est la position de dessin initiale. La position de fin est spécifiée par un déplacement relatif par rapport à la position de début. (Voir la figure 37.)



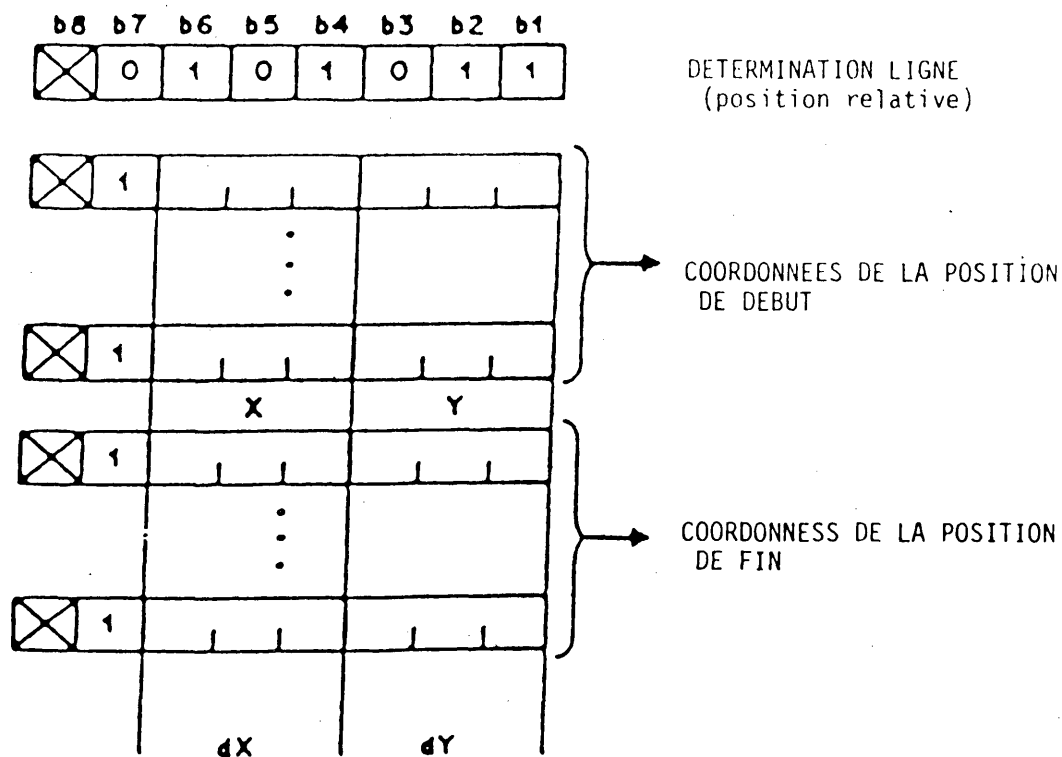
**Figure 37**  
LIGNE (position relative)

**5.3.3.2.4 DÉTERMINATION LIGNE (position absolue).** Les positions de début et de fin sont toutes deux spécifiées par des coordonnées absolues. (Voir la figure 38.) Si plus de 2 opérandes sont présents, des lignes sont tracées entre la première position et la deuxième position, puis entre la troisième et la quatrième, et ainsi de suite.



**Figure 38**  
DÉTERMINATION LIGNE (position absolue)

5.3.3.2.5 DÉTERMINATION LIGNE (position relative). La position de début est spécifiée par des coordonnées absolues, et la position de fin par un déplacement relatif par rapport à la position de début. (Voir la figure 39.)



### 5.3.3.3 ARC

5.3.3.3.1 L'opération de dessin géométrique ARC permet de produire des cercles, des segments de cercle et des tracés curvilignes irréguliers. Dans le cas des cercles et des segments de cercle, un arc est tracé d'une position de début à une position de fin, en passant par une position intermédiaire. Un cercle est tracé lorsque la position de début et la position de fin coïncident; la position intermédiaire définit le diamètre du cercle et se trouve donc à mi-chemin de la position de début et de la position de fin. Un segment de cercle est tracé lorsque la position de début et la position de fin ne coïncident pas.

La position de début est spécifiée explicitement par la commande ARC ou interprétée comme étant la position de dessin actuelle. La position intermédiaire est décrite par un déplacement relatif par rapport à la position de début. La position de fin est spécifiée par un déplacement relatif à partir de la position intermédiaire. Afin de réduire le plus possible les chances d'erreur, il est bon que la position intermédiaire spécifiée sur l'arc se trouve à peu près à mi-chemin de la position de début et de la position de fin.

Si les trois positions de dessin sont colinéaires, une ligne est tracée de la position de début à la position de fin, sauf dans la condition d'erreur où la position intermédiaire ne se trouve pas entre les positions de début et de fin. L'omission de la position de fin suppose que cette position coïncide avec la position de début, et un cercle est alors tracé. À noter que les spécifications d'un arc ne peuvent jamais amener une partie quelconque de cet arc à l'extérieur de l'écran unitaire (voir 5.3.1.1). À la fin du dessin d'un arc, la position de dessin coïncide avec la position de fin.

Un arc peut être représenté par une figure remplie ou par un simple périmètre. Dans le cas des périmètres, la couleur utilisée est la couleur actuelle, la largeur est déterminée par la taille du pel logique et la texture de ligne est spécifiée par la commande TEXTURE. La corde joignant les positions de début et de fin n'est pas considérée comme une partie du périmètre et n'est donc pas tracée.

Dans le cas des arcs remplis, l'aire circonscrite par le périmètre et la corde (y compris la région du périmètre et de la corde tracée par le pel logique) est remplie du motif spécifié par la commande TEXTURE; la couleur actuelle est utilisée pour le remplissage. La largeur de trait de la corde est déterminée par le pel logique, mais la corde n'est pas considérée comme une partie de l'arc et n'est donc pas rehaussée lorsque le mode de rehaussement est sélectionné (voir 5.3.2.4.3 et figure 40).

La spécification de plus de trois positions produit un tracé curviligne irrégulier. La dernière position spécifiée est la position de fin. La position de dessin à la fin d'un tracé curviligne irrégulier est la position de fin. La matérialisation minimale du tracé curviligne irrégulier doit être une série de lignes reliant la position de début, la position intermédiaire et la position de fin du tracé. L'appareil d'affichage peut dessiner un tracé plus lisse, mais la forme de ce tracé et les caractéristiques de l'algorithme utilisé sont liées à la matérialisation. L'algorithme complet du tracé curviligne irrégulier sera

normalisé plus tard. Tous les attributs décrits ci-dessus pour les cercles et segments de cercle (points colinéaires, points à l'extérieur de l'écran unitaire, remplissage et périmètre) s'appliquent aussi aux tracés irréguliers. Dans le cas d'un tracé curviligne irrégulier rempli, le tracé irrégulier et la corde (la ligne qui joint les positions de début et de fin) doivent circonscrire une seule aire, c'est-à-dire qu'aucune partie du tracé ou de la corde ne doit couper une autre partie. Le nombre maximal des positions qui peuvent être utilisées pour décrire un tracé curviligne irrégulier est lié à la matérialisation, mais le nombre minimal doit être 256.

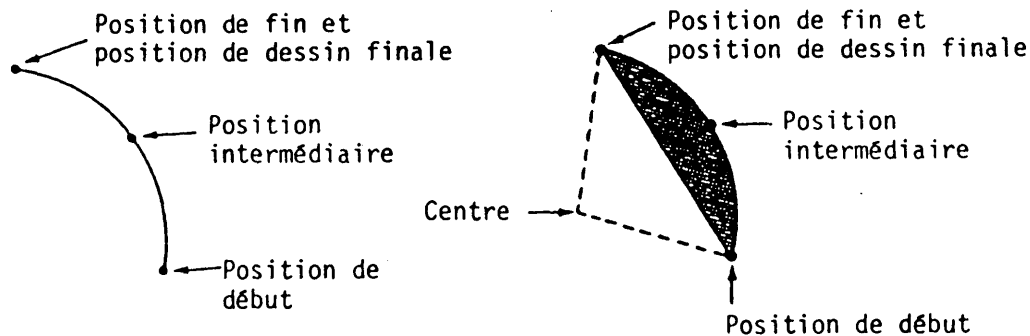


Figure 40  
ARC

**5.3.3.3.2 ARC (périmètre).** La position de début est la position de dessin actuelle, la position intermédiaire est donnée par la première paire de coordonnées spécifiant un déplacement relatif par rapport à la position de début, et la position de fin est donnée par la deuxième paire de coordonnées spécifiant un déplacement relatif par rapport à la position intermédiaire. (Voir la figure 41.) L'arc n'est pas rempli.

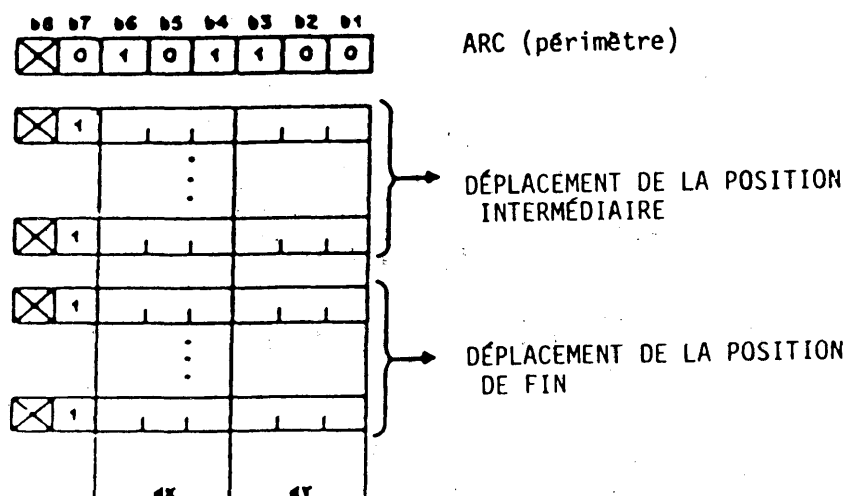
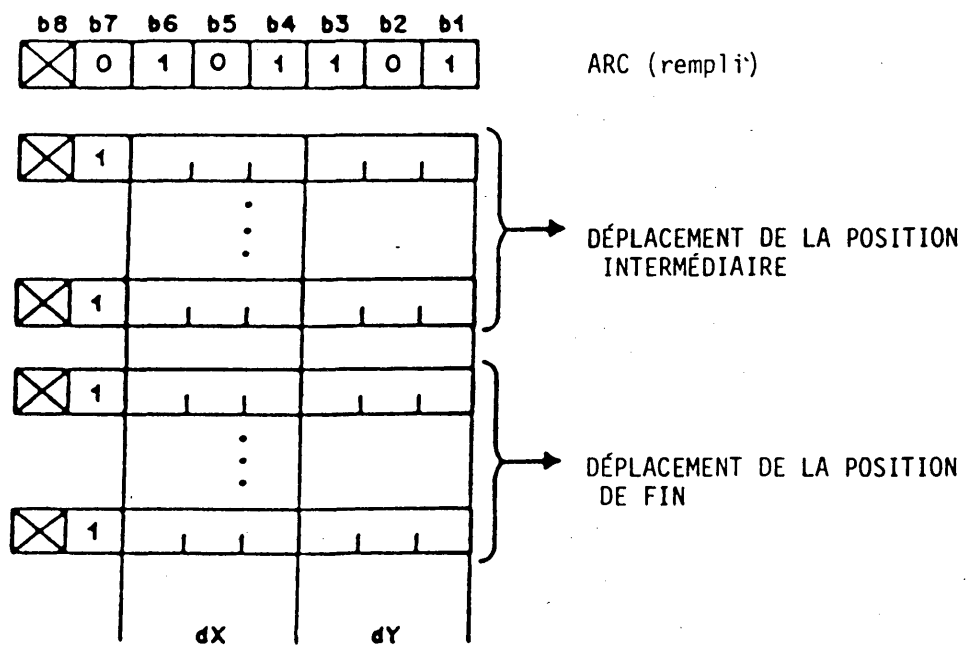


Figure 41  
ARC (périmètre)

**5.3.3.3.3 ARC (rempli).** La position de début est la position de dessin actuelle, la position intermédiaire est donnée par la première paire de coordonnées spécifiant un déplacement relatif par rapport à la position de début, et la position de fin est donnée par la deuxième paire de coordonnées spécifiant un déplacement relatif par rapport à la position intermédiaire. Les positions de début et de fin sont reliées par une corde, et la figure résultante est remplie des couleurs et du motif actuels. (Voir la figure 42.)



**Figure 42**  
ARC (rempli)

**5.3.3.3.4 DÉTERMINATION ARC (périmètre).** La position de début est donnée par la première paire de coordonnées spécifiant une position absolue. La position intermédiaire est donnée par la deuxième paire de coordonnées spécifiant un déplacement relatif par rapport à la position de début, et la position de fin est donnée par la troisième paire de coordonnées spécifiant un déplacement relatif par rapport à la position intermédiaire. (Voir la figure 43.) L'arc n'est pas rempli.

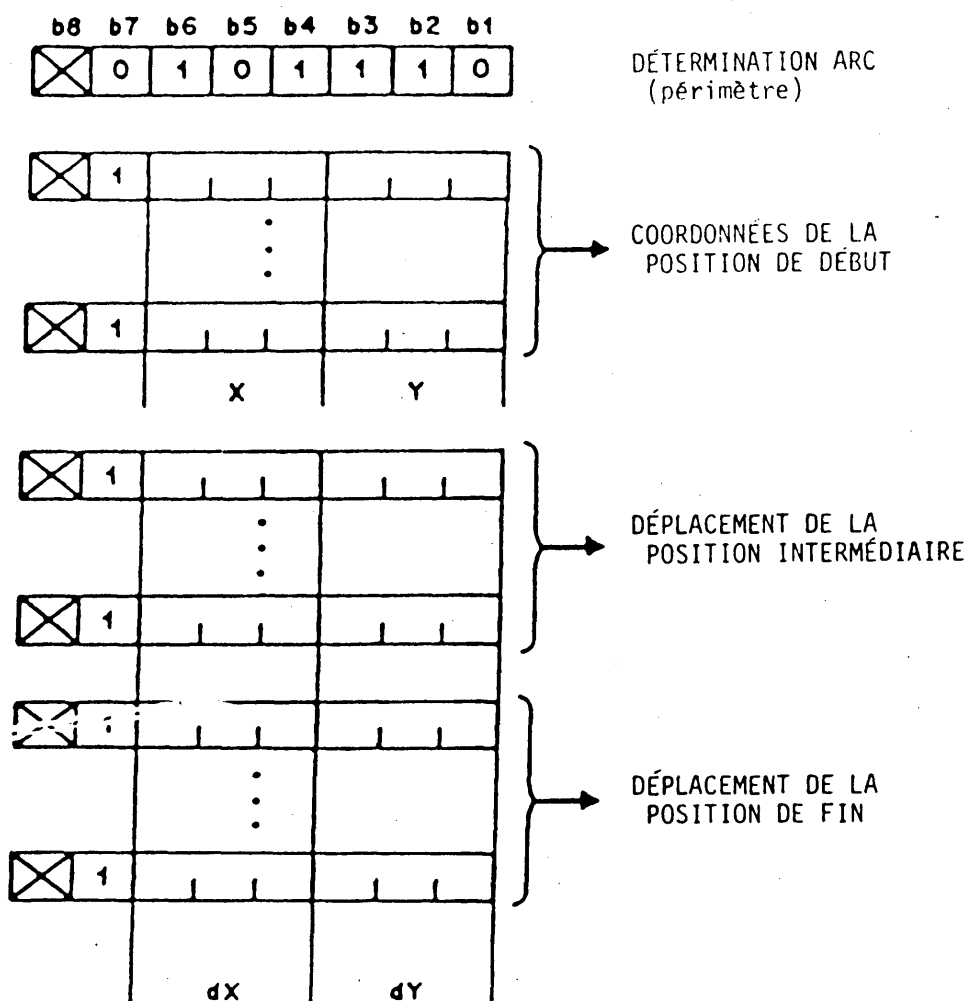
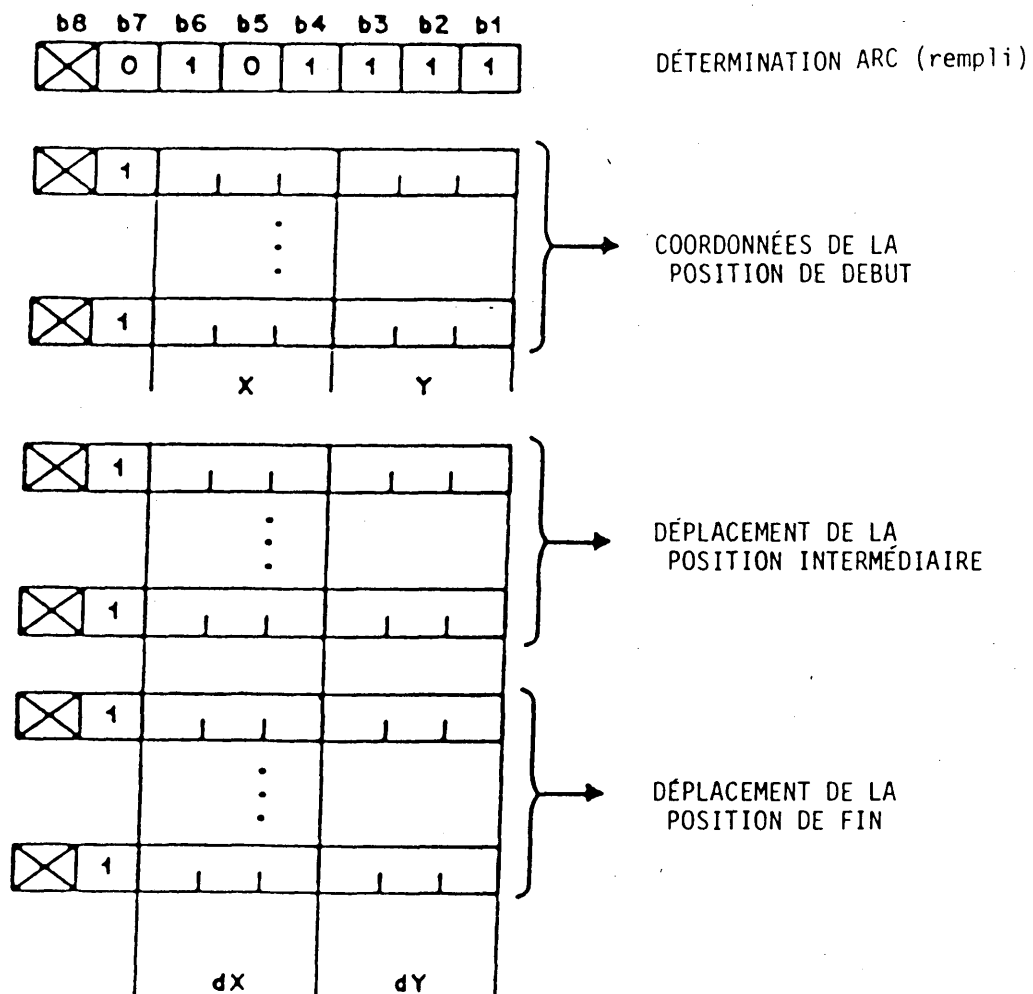


Figure 43  
DÉTERMINATION ARC (périmètre)

**5.3.3.3.5 DÉTERMINATION ARC (rempli).** La position de début est donnée par la première paire de coordonnées spécifiant une position absolue. La position intermédiaire est donnée par la deuxième paire de coordonnées spécifiant un déplacement relatif par rapport à la position de début, et la position de fin est donnée par la troisième paire de coordonnées spécifiant un déplacement relatif par rapport à la position intermédiaire. (Voir la figure 44.) Les positions de début et de fin sont reliées par une corde, et la figure résultante est remplie des couleurs et du motif actuels.



**Figure 44**  
DÉTERMINATION ARC (rempli)



#### 5.3.3.4 RECTANGLE

5.3.3.4.1 La commande RECTANGLE permet de tracer une aire rectangulaire de largeur  $dx$  et de hauteur  $dy$ . La position de début est spécifiée explicitement par la commande RECTANGLE, ou encore interprétée comme étant la position de dessin actuelle. À la fin du dessin d'un rectangle, la position de dessin est la position de début modifiée seulement sur l'axe des  $x$ , par la valeur du déplacement  $dx$ .

Un RECTANGLE peut être représenté par une figure remplie ou par un simple périmètre. Dans le cas des périmètres, la couleur utilisée est la couleur actuelle, la largeur de ligne est déterminée par la taille du pel logique et la texture de ligne est spécifiée par la commande TEXTURE. Dans le cas des rectangles remplis, les couleurs actuelles et le motif spécifié par la commande TEXTURE sont utilisés à l'intérieur de l'aire circonscrite par le périmètre (y compris la partie du périmètre tracée par le pel logique), et le périmètre est rehaussé si le mode de rehaussement est sélectionné (voir 5.3.2.4.3).

La commande RECTANGLE peut servir à tracer un histogramme à partir d'un tableau de nombres représentant des déplacements relatifs  $dy$  et  $dx$ , de la même façon que les commandes POINT et LIGNE sont employées pour dessiner des diagrammes (voir la figure 45).

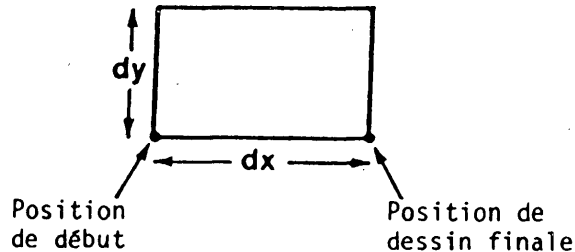


Figure 45  
RECTANGLE

5.3.3.4.2 RECTANGLE (périmètre). La position de début est la position de dessin actuelle, et la largeur ainsi que la hauteur (dx, dy) sont données par la première paire de coordonnées. (Voir la figure 46.) Le rectangle n'est pas rempli.

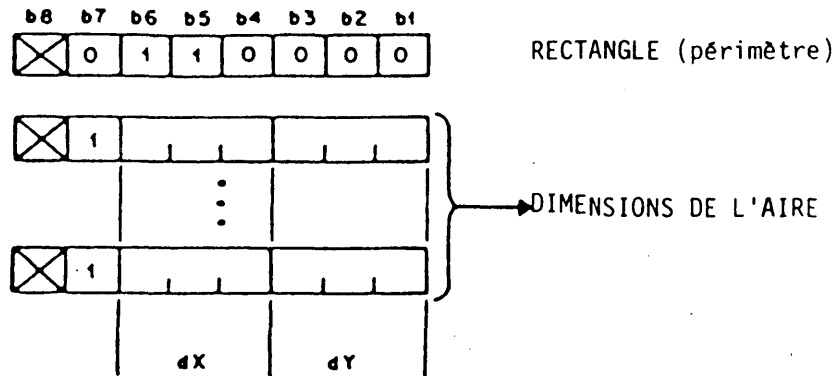


Figure 46  
RECTANGLE (périmètre)

5.3.3.4.3 RECTANGLE (rempli). La position de début est la position de dessin actuelle, et la largeur ainsi que la hauteur (dx, dy) sont données par la première paire de coordonnées. (Voir la figure 47.) Le rectangle est rempli des couleurs et du motif actuels.

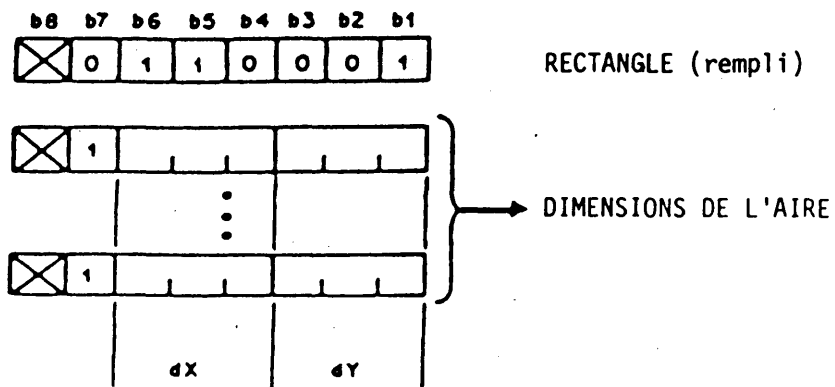
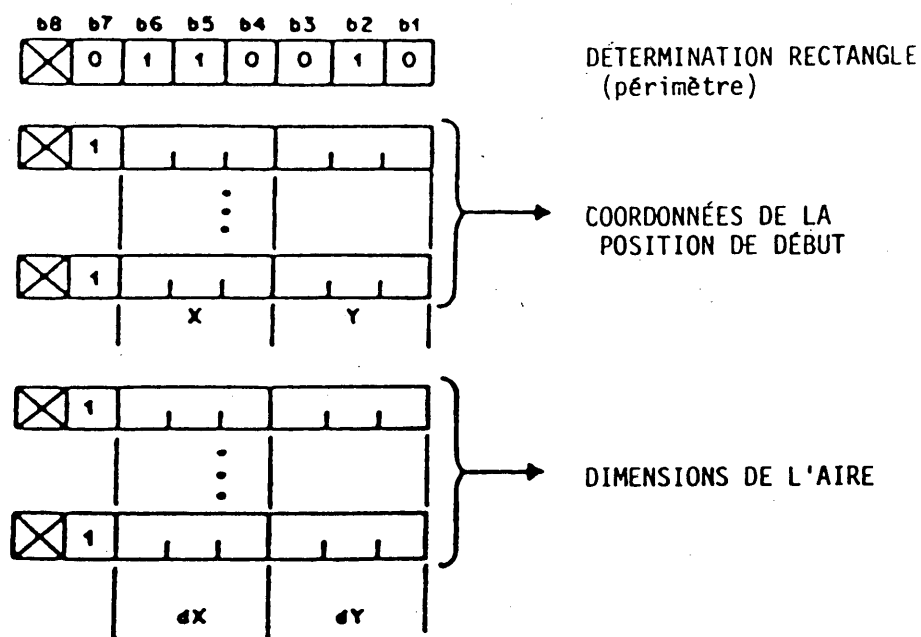


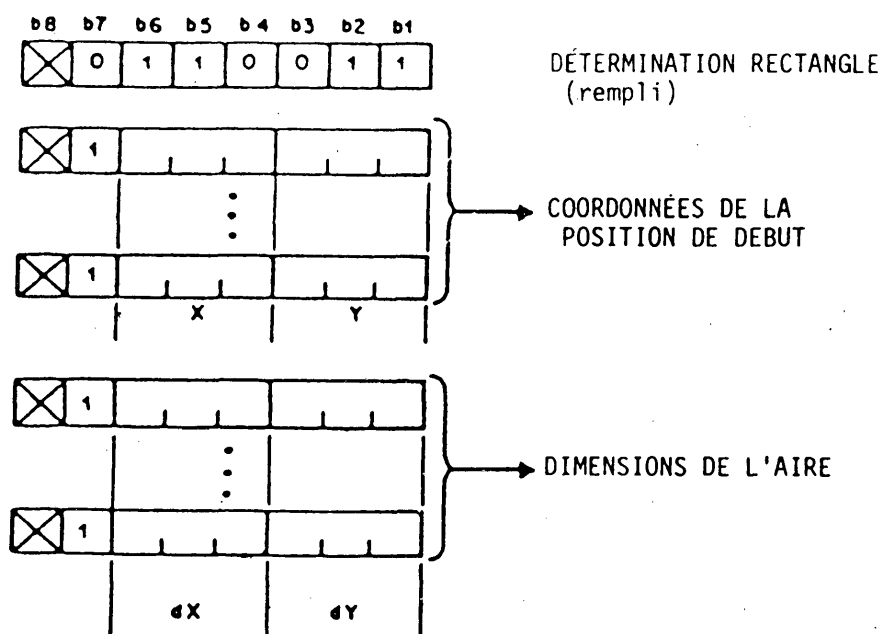
Figure 47  
RECTANGLE (rempli)

**5.3.3.4.4 DÉTERMINATION RECTANGLE (périmètre).** La position de début est spécifiée de façon absolue par la première paire de coordonnées, et la largeur ainsi que la hauteur (dx, dy) sont données par la deuxième paire de coordonnées. (Voir la figure 48.) Le rectangle n'est pas rempli.



**Figure 48**  
DÉTERMINATION RECTANGLE (périmètre)

**5.3.3.4.5 DÉTERMINATION RECTANGLE (rempli).** La position de début est spécifiée de façon absolue par la première paire de coordonnées, et la largeur ainsi que la hauteur (dx, dy) sont données par la deuxième paire de coordonnées. (Voir la figure 49.) Le rectangle est rempli des couleurs et du motif actuels.



**Figure 49**  
DÉTERMINATION RECTANGLE (rempli)

### 5.3.3.5 POLYGONE

5.3.3.5.1 La commande POLYGONE permet de tracer un polygone quelconque à partir de ses sommets. Un POLYGONE est défini par des paires de coordonnées correspondant à ces sommets. La position de début est précisée explicitement par la commande POLYGONE, ou correspond à la position de dessin actuelle. Chaque paire de coordonnées (dx, dy) représente un déplacement relatif par rapport au sommet précédent (un déplacement relatif de valeur 0 est sans effet). La fermeture entre la position de début et le dernier sommet est considérée comme implicite, de sorte qu'à la fin du tracé la position de dessin coïncide avec la position de début. (Voir la figure 50.)

Un polygone peut être représenté par une figure remplie ou par un simple périmètre. Dans le cas des périmètres, les couleurs actuelles sont utilisées, la largeur de ligne est déterminée par la taille du pel logique et la texture de ligne est spécifiée par la commande TEXTURE. Dans le cas des polygones remplis, les couleurs actuelles et le motif spécifié par la commande TEXTURE sont utilisés à l'intérieur de l'aire (y compris le périmètre tracé par le pel logique), et le périmètre est rehaussé si le mode de rehaussement a été sélectionné (voir 5.3.2.4.3).

Un polygone ne doit circonscrire qu'une seule aire, c'est-à-dire qu'aucune ligne joignant deux sommets consécutifs ne peut couper une autre ligne joignant aussi deux sommets consécutifs.

Le nombre de sommets décrivant un polygone est déterminé par le nombre de données suivant le code d'opération POLYGONE. Le nombre maximal de sommets pouvant décrire un polygone est lié à la matérialisation mais doit comporter au moins 256 sommets.

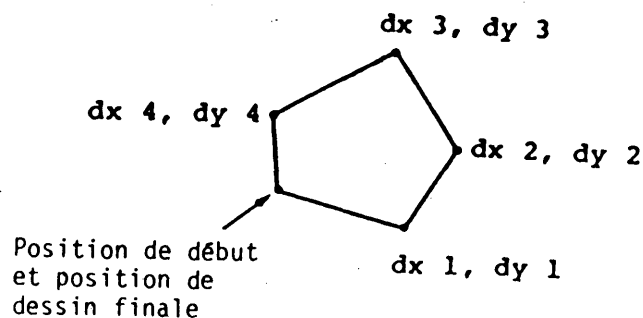


Figure 50  
POLYGONE

5.3.3.5.2 POLYGONE (périmètre). La position de début correspond à la position de dessin actuelle, et les coordonnées des sommets subséquents sont spécifiées sous forme de déplacement relatif par rapport au sommet qui les précède. (Voir la figure 5.) Le polygone n'est pas rempli.

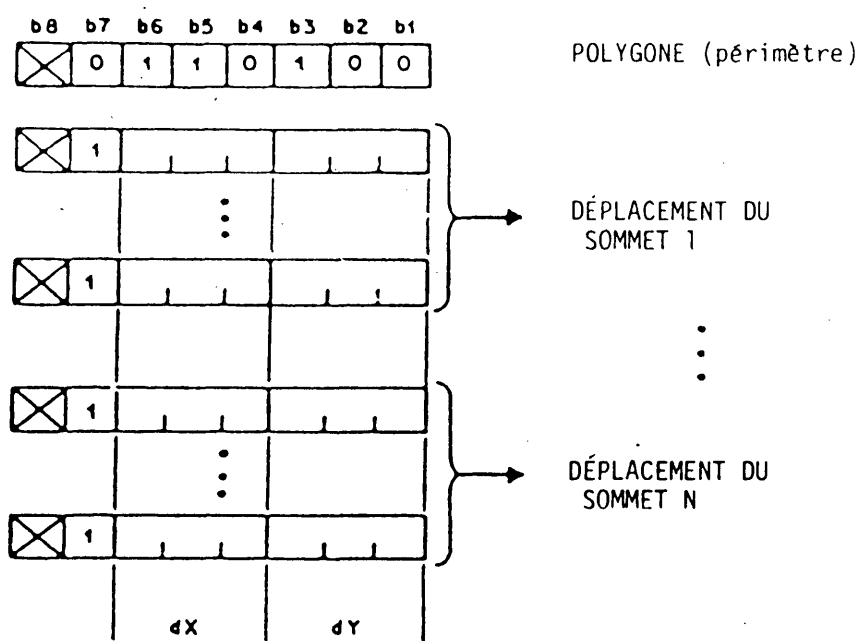


Figure 51  
POLYGONE (périmètre)

5.3.3.5.3 POLYGONE (rempli). La position de début correspond à la position de dessin actuelle, et les coordonnées des sommets subséquents sont spécifiées sous forme de déplacement relatif par rapport au sommet qui les précède. (Voir figure 52.) Le polygone est rempli des couleurs et du motif actuels.

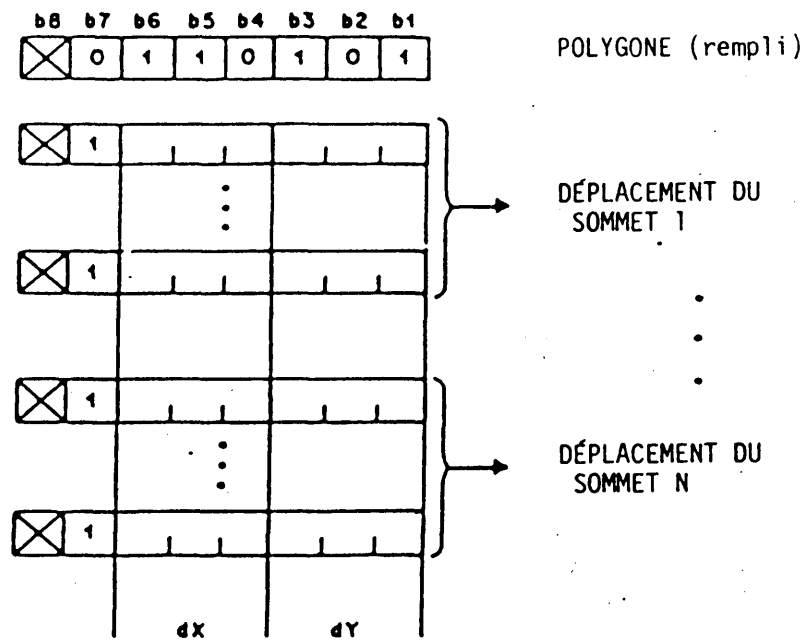
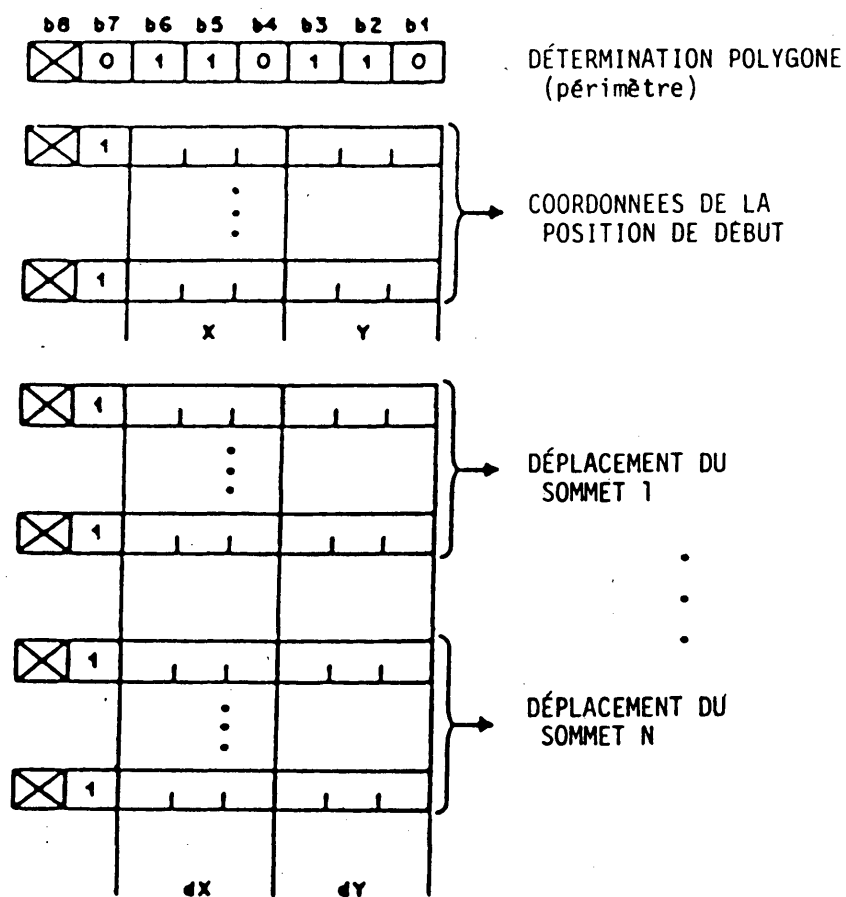


Figure 52

POLYGONE (rempli)

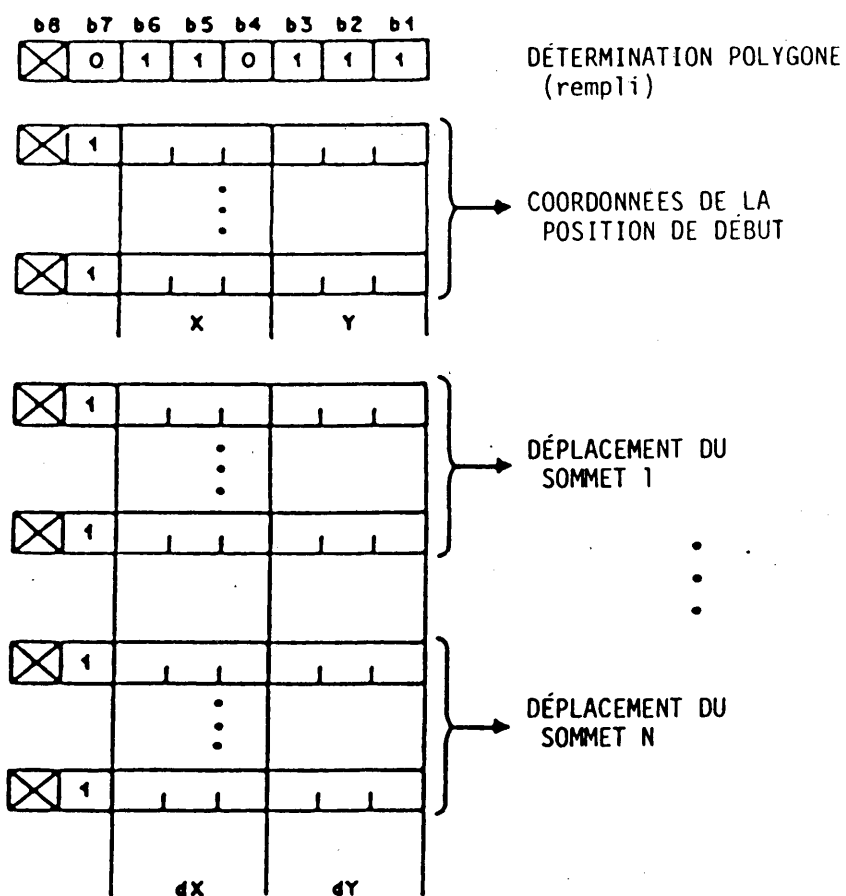
**5.3.3.5.4 DÉTERMINATION POLYGONE (périmètre).** La position de début est spécifiée de façon absolue par la première paire de coordonnées, et les coordonnées des sommets subséquents sont spécifiées sous forme de déplacement relatif par rapport au sommet qui les précède. (Voir la figure 53.) Le polygone n'est pas rempli.



**Figure 53**  
DÉTERMINATION POLYGONE (périmètre)



**5.3.3.5.5. DÉTERMINATION POLYGONE (rempli).** La position de début est spécifiée de façon absolue par la première paire de coordonnées, et les coordonnées des sommets subséquents sont spécifiées sous forme de déplacement relatif par rapport au sommet qui les précède. (Voir la figure 54.) Le polygone est rempli des couleurs et du motif actuels.



**Figure 54**  
DÉTERMINATION POLYGONE (rempli)

### 5.3.3.6 INCRÉMENTATION

5.3.3.6.1 La commande INCRÉMENTATION permet de décrire des images complexes de façon condensée. Il existe quatre codes d'opération INCRÉMENTATION: CHAMP, INCRÉMENTATION POINT, INCRÉMENTATION LIGNE et INCRÉMENTATION POLYGONE (rempli).

L'image peut être de nature photographique (CHAMP et INCRÉMENTATION POINT), ou elle peut être constituée de lignes complètes comme dans le cas des signatures (INCRÉMENTATION LIGNE), ou encore de polygones remplis comme dans le cas des logotypes ou autres symboles (INCRÉMENTATION POLYGONE).

5.3.3.6.2 CHAMP. La commande CHAMP sert à définir le champ actif utilisé pour les textes en colonnes, les champs sans protection et la commande INCRÉMENTATION POINT. L'origine du champ est spécifiée de façon absolue par la première paire de coordonnées (X, Y). La paire de coordonnées suivante donne les dimensions du champ, c'est-à-dire la largeur et la hauteur (dx, dy). (Voir la figure 55.) À noter que les coordonnées dx et dy peuvent être séparément ou simultanément positives ou négatives, ce qui permet de placer l'origine à n'importe quel des quatre coins du champ. La position de dessin se trouve à l'origine du champ après l'exécution de la commande CHAMP. Un seul champ actif peut être défini à la fois, c'est-à-dire que l'exécution de la commande CHAMP supprime tout champ actif précédent. Si le code d'opération CHAMP n'est pas suivi par des multiplets de données, le champ actif correspond à l'écran unitaire et l'origine est (0,0). Si le code d'opération CHAMP est suivi d'un seul opérande, cet opérande spécifie les dimensions du champ, et l'origine est la position de dessin actuelle. Les multiplets de données numériques suivant le deuxième opérande seront normalisés plus tard et ils doivent rester sans effet. Le champ actif par défaut est l'écran unitaire.

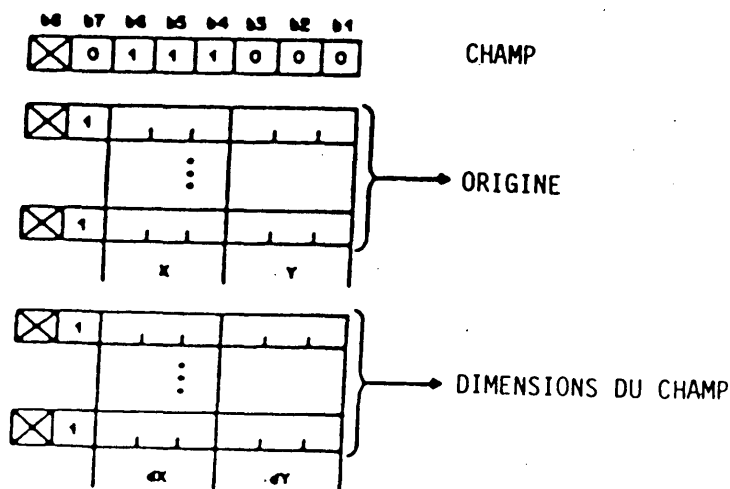


Figure 55  
CHAMP

**5.3.3.6.3 INCRÉMENTATION POINT.** Grâce à la commande INCRÉMENTATION POINT, il est possible de décrire une image par une chaîne de spécifications de couleurs qui sont appliquées dans le champ actif à la manière d'une trame séquentielle. Ces spécifications de couleurs sont contenues dans une chaîne opérande. En mode de couleur 0, ces spécifications sont interprétées comme des couleurs réelles. Dans les modes de couleur 1 et 2, ces spécifications sont interprétées comme des adresses de la carte des couleurs. L'algorithme de construction de l'image se déroule de la façon suivante:

- 1) L'opération commence à la position de dessin actuelle.
- 2) Si aucune partie du pel logique ne déborde le champ actif, la première couleur obtenue de la chaîne opérande de chaîne est appliquée aux positions de la mémoire d'affichage correspondant aux pixels visés par le pel logique. Si un pixel est visé par le pel logique associé à la position de dessin durant plus d'une opération d'application, il retient la couleur qui vient d'être appliquée. La position de dessin est ensuite automatiquement décalée dans le sens X d'une distance égale à la largeur (dx) du pel logique. À noter que si la valeur dx est positive la position de dessin est décalée vers la droite; si elle est négative, vers la gauche. La couleur suivante est ensuite obtenue et le processus se répète.
- 3) Si une partie quelconque du pel logique déborde le champ actif, tous les autres bits du multiplet de la chaîne opérande actuellement interprétée sont omis, même s'il reste un nombre suffisant de bits pour constituer une spécification de couleur complète. L'interprétation reprend au premier bit (c.-à-d. b6) du multiplet complet suivant. Si les multiplets de données numériques sont épuisés, l'opération prend fin; dans le cas contraire, la position de dessin est ramenée à la limite opposée. Si le décalage de la position de dessin dans le sens Y d'une distance égale à la hauteur (dy) du pel logique entraîne le débordement d'une partie quelconque du pel logique à l'extérieur du champ actif, la valeur Y est maintenue constante et l'image se trouvant dans la plage de dessin définie par le champ actif défile dans le sens opposé, c.-à-d. d'une distance égale à -dy; dans le cas contraire, la position de dessin est décalée dans le sens Y d'une distance égale à la hauteur (dy) du pel logique. À noter que si la valeur dy est positive la position de dessin est décalée vers le haut; si elle est négative, vers le bas. Si l'opération n'est pas terminée, le processus se poursuit à l'étape 2. Si l'opération est terminée, la position de dessin est ramenée à l'origine du champ actif.

Le code d'opération INCRÉMENTATION POINT et ses opérandes sont illustrés sur la figure 56. La figure 57 montre une image obtenue par la commande INCRÉMENTATION POINT.

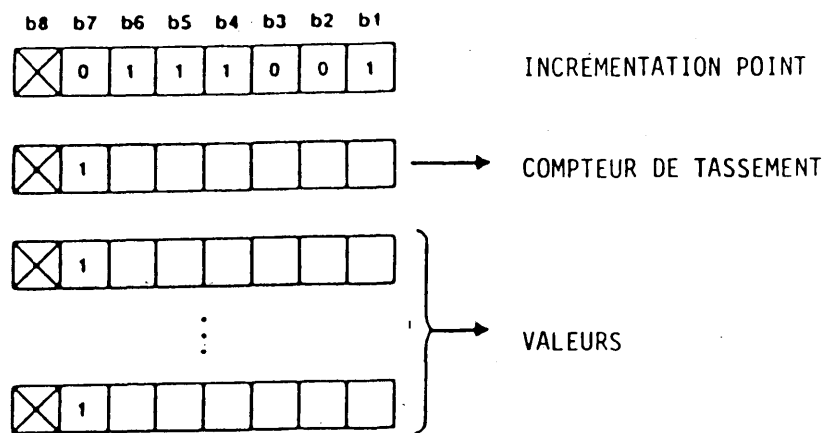


Figure 56  
INCRÉMENTATION POINT



Figure 57  
Exemple d'image obtenue par la commande INCRÉMENTATION POINT

La commande INCRÉMENTATION POINT comporte deux opérandes. Le premier est formé par un seul multiplet de format fixe qui décrit le compteur de tassement. Il s'agit d'un nombre entier sans signe qui détermine le nombre de bits consécutifs à prélever de la chaîne opérande pour produire une seule spécification de couleur. La plage des valeurs du compteur de tassement s'étend de 1 à 48 inclusivement. La valeur 0 et toutes les valeurs supérieures à 48 seront normalisées plus tard; toute commande INCRÉMENTATION POINT comportant des compteurs de tassement portant ces valeurs doit être exécutée comme une opération vide. Le second opérande est une chaîne de longueur indéterminée qui indique les spécifications de couleur, mémorisées de façon séquentielle sans égard aux limites des multipléts, du bit de poids fort (b6) au bit de poids faible (b1), dans les champs de données numériques. En mode de couleur 0, ces spécifications de couleur sont traitées comme des couleurs réelles, c.-à-d. qu'un nombre de bits égal au compte de tassement est utilisé pour définir la couleur appliquée à chaque opération de dessin. Comme dans le cas de la spécification à valeurs multiples, les bits sont organisés en triplets qui s'interprètent dans l'ordre V, R, B. À noter que la spécification d'une seule couleur peut contenir des triplets multiples selon le compte de tassement, et dans ce cas le premier triplet contient les bits de poids fort et le dernier triplet, les bits de poids faible des trois primaires. Par exemple, avec un compte de tassement égal à 6, la spécification de couleur pour chaque opération de dessin prend la forme VRBVRB, chaque primaire étant spécifiée à deux bits de précision. Si le compte de tassement n'est pas un multiple entier de trois, les spécifications des trois couleurs primaires ne présenteront pas la même précision. Par exemple, si le compte de tassement est égal à 4, la spécification de couleur pour chaque opération de dessin prend la forme VRBV. À noter que ces spécifications sont concaténées dans la chaîne opérande sans égard aux limites des multipléts. Dans notre exemple, par conséquent, les bits de la chaîne opérande prennent la forme VRBVVRBV ...

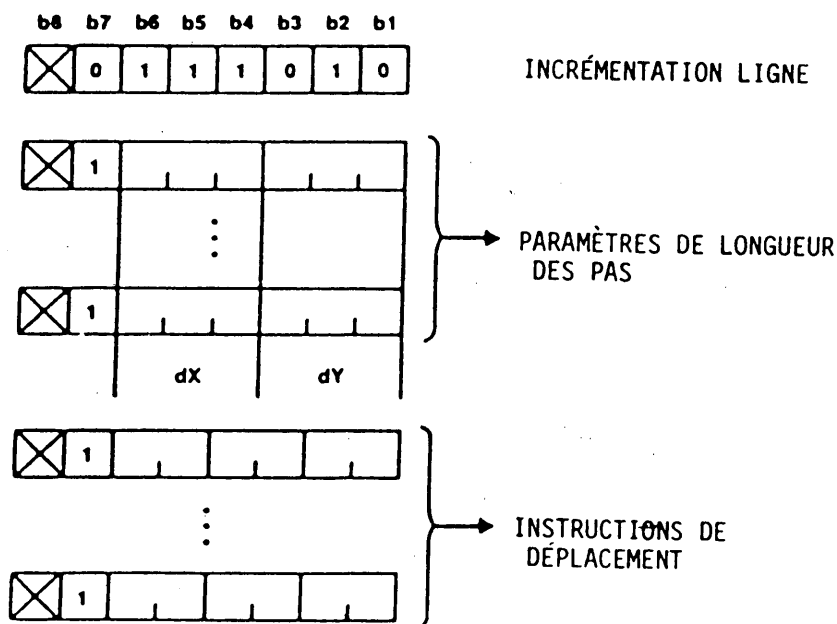
Dans les modes de couleur 1 et 2, ces spécifications de couleur sont des nombres ordinaux, c.-à-d. des adresses de la carte des couleurs auxquelles la vraie valeur de couleur à déjà été ou sera chargée. Une fois de plus, un nombre de bits égal au compte de tassement est utilisé pour définir la couleur appliquée à chaque opération de dessin. Ces spécifications sont concaténées dans la chaîne opérande sans égard aux limites des multipléts.

Il est nécessaire, à partir de la relation entre les dimensions du pel logique et les dimensions du pixel physique d'un appareil d'affichage particulier, de procéder avant l'exécution à une conversion d'échelle des dimensions du pel logique et des dimensions du champ actif pour éviter certains types de déformation (déformation oblique) résultant d'un désaccord. Lorsque la position de dessin se trouve dans un champ actif de ce genre, l'emplacement relatif de la position de dessin à l'intérieur du champ mis à l'échelle ne doit pas changer après la conversion d'échelle. Le but de ce type de conversion est de transformer les dimensions du pel logique en multiples entiers ou en fractions entières des dimensions du pixel physique correspondant. (Les dimensions du champ actif doivent être mises à l'échelle de façon équivalente pour empêcher la déformation oblique de l'image.) Les conditions ci-dessous sont nécessaires: (1) les dimensions du pel logique et les dimensions du champ actif sont ramenées à la valeur précédant la conversion d'échelle après l'exécution de la commande

INCRÉMENTATION POINT; (2) l'image résultante doit se trouver dans les limites du champ actif original; et (3) la matérialisation doit assurer que la déformation oblique ne se produit pas, quelles que soient les précisions des dimensions du pel logique et du champ spécifiées par la commande DOMAINE.

Si, à la réception d'une commande INCRÉMENTATION POINT, une partie quelconque du pel logique indiqué par la position de dessin initiale se trouve à l'extérieur du champ actif, la commande est considérée comme erronée et est exécutée comme une opération vide. Si, à la réception d'une commande INCRÉMENTATION POINT, l'une des deux dimensions ou les deux dimensions du pel logique sont égales à 0, ces dimensions prennent, pour la durée de la commande seulement, la plus petite valeur positive spécifiable dans le domaine actuel. (Par exemple, si la longueur de l'opérande à valeurs multiples actuel est 3 multiplats, la plus petite valeur spécifiable est +0.00000001, c.-à-d. 1/256.)

**5.3.3.6.4 INCRÉMENTATION LIGNE.** La commande INCRÉMENTATION LIGNE permet de décrire de façon condensée une image constituée d'une série de courts segments de ligne de la couleur (ou des couleurs) et de la texture actuelles. (Voir les figures 58 et 59.)



**Figure 58**  
INCRÉMENTATION LIGNE

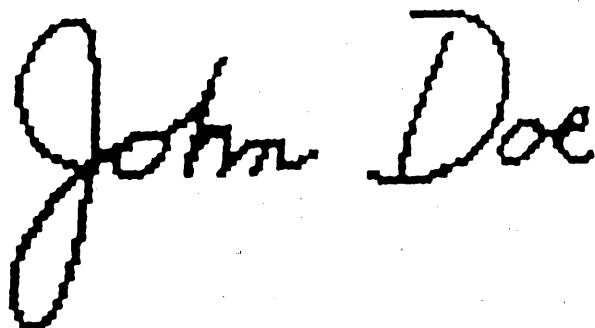


Figure 59

Exemple d'image obtenue par la commande INCRÉMENTATION LIGNE

Le premier opérande à valeurs multiples indique la longueur des pas sous forme de paramètres de déplacement avec signe, dx et dy.

Le dernier bloc de données est une chaîne opérande qui indique les valeurs de déplacement par un nombre indéfini de multipléts comportant chacun trois doublets, b6 à b1, dans le champ de données numériques. L'interprétation de ces doublets est donnée au tableau 16.

Tableau 16

Valeurs de déplacement d'incrémentatation ligne

Valeur du doublet		Interprétation
0	0	Interpréter le doublet suivant comme une instruction "modification de paramètre" (voir tableau 17).
0	1	Faire avancer la position de dessin d'une distance dx sur l'axe des x, et tracer facultativement une ligne.
1	0	Faire avancer la position de dessin d'une distance dy sur l'axe des y, et tracer facultativement une ligne.
1	1	Faire avancer la position de dessin d'une distance dx sur l'axe des x et dy sur l'axe des y, et tracer facultativement une ligne.

Lorsque le drapeau de dessin est au niveau vrai, une ligne de la texture et de la couleur actuelles est tracée entre la position de dessin actuelle (après la progression d'un pas) et la position de dessin précédente (avant la progression d'un pas), chaque fois qu'il y a progression. Lorsque le drapeau de dessin est au niveau faux, aucune ligne n'est tracée après la progression. Lorsqu'un doublet a la valeur (0, 0) (tableau 16), le doublet suivant est interprété comme une instruction "modification de paramètre", comme l'indique le tableau 17. Le doublet subséquent est interprété comme une opération de progression d'un pas (tableau 16). Le drapeau de dessin est initialement au niveau vrai chaque fois que se présente le code d'opération INCRÉMENTATION LIGNE. À la fin de la chaîne opérante, la position de dessin reste à la position de dessin finale.

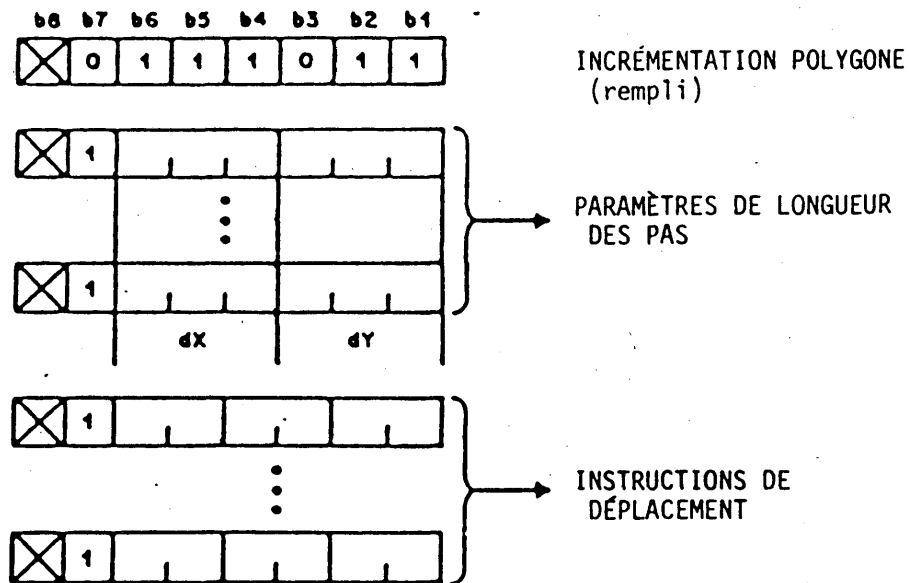
Tableau 17

Instructions de modification de paramètre  
d'incrémentatation ligne

Valeur du doublet		Instruction de modification de paramètre
0	0	Changer l'état du drapeau de dessin (de vrai à faux ou de faux à vrai)
0	1	Changer le signe de dx.
1	0	Changer le signe de dy.
1	1	Changer le signe de dx et dy.



**5.3.3.6.5 INCRÉMENTATION POLYGONE (rempli).** La commande INCRÉMENTATION POLYGONE permet de décrire de façon condensée un polygone rempli tracé avec une série de courts segments de ligne. (Voir les figures 60 et 61.) L'aire circonscrite par le périmètre (y compris la région du périmètre tracée par le pel logique) est remplie de la couleur actuelle (ou des couleurs actuelles) et du motif spécifié par la commande TEXTURE, et le périmètre est rehaussé si le mode de rehaussement a été sélectionné (voir 5.3.2.4.3).



**Figure 60**  
INCRÉMENTATION POLYGONE (rempli)



Figure 61

Exemple d'image obtenue par la commande **INCRÉMENTATION POLYGONE**

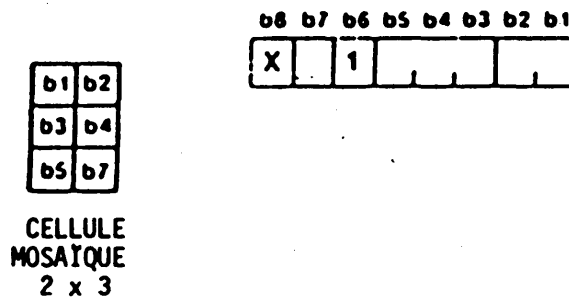
L'interprétation de l'opérande suivant le code d'opération se fait exactement comme dans le cas du code d'opération **INCRÉMENTATION LIGNE**, avec les exceptions suivantes:

- 1) Le drapeau de dessin est toujours au niveau vrai.
- 2) En présence d'une instruction "modification de paramètre" (tableau 17) visant à changer l'état du drapeau de dessin (0, 0), cette instruction est considérée comme vide et le doublet suivant est interprété comme l'instruction "modification de paramètre".
- 3) La position de dessin finale est considérée implicitement comme étant la position de dessin initiale.
- 4) La figure résultante est remplie de la couleur et du motif actuels, compte tenu de l'attribut de rehaussement.

A noter que la commande **INCRÉMENTATION POLYGONE** ne doit s'appliquer qu'à une seule aire (tout comme la commande **POLYGONE**, alinéa 5.3.3.5).

**5.4 Jeu mosaïque.** La figure 63 montre l'assignation des caractères du jeu mosaïque. Ce jeu comporte 65 caractères, c.-à-d. 65 cellules mosaïques 2 x 3, y compris une seconde cellule mosaïque pleine à la position 5/15. Les positions de caractère inutilisées seront normalisées plus tard; elles doivent être affichées comme un ESPACE. Les caractères mosaïques peuvent être affichés dans deux modes, le mode contigu et le mode distinct, selon le mode de soulignement décrit en 6.2.7.15. En mode contigu, les six éléments mosaïques formant chaque cellule de caractère doivent occuper tout le champ de caractère donné, quelles que soient les dimensions. En mode distinct, chaque élément mosaïque affiché est réduit dans le sens horizontal de la valeur absolue de la largeur (dx) du pel logique, et dans le sens vertical de la valeur absolue de la hauteur (dy) du pel logique. (Le pel logique est décrit en 5.3.2.2.) Les éléments mosaïques réduits, dans ce cas, sont justifiés à gauche et au bas de l'aire normale de l'élément, le reste de l'aire étant considéré comme fond. Si l'une ou l'autre dimension du pel logique a une valeur égale ou supérieure à la dimension correspondante de l'élément mosaïque, l'aire affichée est réduite à zéro. Le symbole graphique de la position 2/0 du jeu mosaïque ne peut pas être souligné. Les caractères mosaïques (en mode distinct ou contigu) ne peuvent pas faire l'objet d'un espacement proportionnel.

L'algorithme qui détermine le choix des sous-éléments provient de combinaisons de bits tirées du tableau de code de référence. (Voir la figure 62.)



















































































































**Figure 62**

#### Codage mosaïque des sous-éléments

La combinaison de code 5/15 (1011111) commande l'entrée en jeu de tous les sous-éléments. La figure 63 illustre le tableau de code du système mosaïque.

La séquence utilisée pour désigner le jeu mosaïque est ESC I 7/13, la lettre I représentant la position 2/9 ou 2/13 pour indiquer G1, 2/10 ou 2/14 pour indiquer G2, ou 2/11 ou 2/15 pour indiquer G3 (voir 4.3).

					10	11	12	13	14	15	
					b7	0	0	1	1	1	1
					b6	1	1	0	0	1	1
					b5	0	1	0	1	0	1
					COLONNE	2	3	4	5	6	7
b4	b3	b2	b1	RANGÉE							
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							

Note: Les rectangle entourant les caractères sont tracés aux fins d'illustrations; ils ne font pas partie intégrante des symboles graphiques.

Figure 63

Jeu mosaïque

**5.5 Jeu de macro-instructions.** Les macro-instructions permettent de coder des séquences de données du protocole de la couche présentation à exécuter sur commande. Un macro-code est constitué d'une chaîne arbitraire de données de caractères de code de la couche présentation mise en tampon localement et identifiée par un code venant du jeu G de macro-instructions. La macro-instruction joue par la suite le rôle d'un substitut de toute la chaîne de caractères constituant ce macro-code particulier. Il est possible de définir simultanément jusqu'à 96 macro-codes appelés par 96 macro-instructions (voir 6.2.2). Pour utiliser un macro-code, il suffit de désigner le jeu de macro-instructions parmi les jeux G, d'appeler le jeu de macro-instructions dans le tableau effectif et de transmettre la macro-instruction désirée. Une macro-instruction peut être emboîtée dans une macro-code. C'est au niveau de la couche application qu'il importe d'éviter les bouclages sans fin.

Toute macro-instruction (définie par DÉF MACRO, DÉFP MACRO ou DÉFT MACRO, alinéa 6.2.2) peut être associée à un mécanisme d'entrée de l'utilisateur, comme une touche de fonction, pour permettre l'exécution ou la transmission du code par l'utilisateur. Le nombre de macro-instructions associées à un tel mécanisme est lié à la matérialisation.

La séquence utilisée pour désigner le jeu de macro-instructions est ESC I 7/10, la lettre I représentant 2/9 ou 2/13 pour indiquer G1, 2/10 ou 2/14 pour indiquer G2, ou 2/11 ou 2/15 pour indiquer G3 (voir 4.3).

**5.6 Jeu de caractères dynamiquement redéfinissables (JCDR).** Au contraire des autres jeux de caractères, dont les définitions de configuration sont mémorisées en permanence dans l'appareil récepteur et ne peuvent pas être modifiées par le prestataire d'information, le jeu de caractères dynamiquement redéfinissables (JCDR) désigné par une séquence d'échappement à trois caractères permet de transférer et d'utiliser un maximum de 96 configurations, de la même façon que les jeux primaire, supplémentaire et mosaïque, définies par l'utilisateur. Au moment de l'affichage, ces jeux peuvent recevoir les mêmes attributs que le texte alphanumérique. La technique de téléchargement des configurations est décrite en 6.2.3. Si un caractère JCDR n'a pas été défini par la commande DÉF JCDR, il doit être affiché comme un ESPACE.

La séquence utilisée pour désigner le jeu JCDR est ESC I 7/11, la lettre I représentant 2/9 ou 2/13 pour indiquer G1, 2/10 ou 2/14 pour indiquer G2, ou 2/11 ou 2/15 pour indiquer G3 (voir 4.3).

## **6. Codage des jeux C**

### **6.1 Jeu de commande C0**

**6.1.1** Le présent alinéa décrit le jeu de commande C0 (voir figure 64) qui occupe les colonnes 0 et 1 des tableaux effectifs à 7 et 8 éléments. Les fonctions sont décrites ci-dessous.

#### **6.1.2 Caractères de commande de mise en page**

**6.1.2.1 POSITION ACTIVE VERS L'ARRIÈRE (APB).** Ce caractère (0/8) (retour arrière) sert à déplacer le curseur d'une distance égale à l'espace entre les caractères parallèlement au trajet de caractère, dans le sens opposé au trajet (c'est-à-dire à 180° du sens du trajet). Lorsque le déplacement amènerait une partie quelconque du champ de caractère correspondant au-delà de la plage d'affichage (ou au-delà du champ actif si le champ de caractère se trouvait entièrement dans les limites du champ actif immédiatement avant le déplacement), le curseur est plutôt positionné à l'extrémité opposée (dans le sens du trajet de caractère de la plage d'affichage (ou du champ actif), et une commande automatique de position active vers le haut (APU) est exécutée.

**6.1.2.2 POSITION ACTIVE VERS L'AVANT (APF).** Ce caractère (0/9) (tabulation horizontale) sert à déplacer le curseur d'une distance égale à l'espace entre les caractères parallèlement au trajet de caractère, dans le sens du trajet. Lorsque le déplacement amènerait une partie quelconque du champ de caractère correspondant au-delà de la plage d'affichage (ou au-delà du champ actif si le champ de caractère se trouvait entièrement dans les limites du champ actif immédiatement avant le déplacement), le curseur est plutôt positionné à l'extrémité opposée (dans le sens du trajet de caractère de la plage d'affichage (ou du champ actif) et une commande automatique de position active vers le bas (APD) est exécutée.

**6.1.2.3 POSITION ACTIVE VERS LE BAS (APD).** Ce caractère (0/10) (interligne) sert à déplacer le curseur d'une distance égale à l'espace entre les rangées, perpendiculairement au trajet de caractère, dans un sens perpendiculaire au trajet (-90 degrés). Lorsque le déplacement amènerait une partie quelconque du champ de caractère correspondant au-delà de la plage d'affichage (ou au delà du champ actif si le champ de caractère se trouvait entièrement dans les limites du champs actif immédiatement avant le déplacement), des opérations spéciales sont exécutées en fonction du mode de défilement (avec ou sans défilement) (voir 6.2.7.13 et 6.2.7.14).

				b <sub>7</sub>	0	0
				b <sub>6</sub>	0	0
				b <sub>5</sub>	0	1
				COLONNE	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	RANGÉE		
0	0	0	0	0	NUL	DLE
0	0	0	1	1	SOH	DC1
0	0	1	0	2	STX	DC2
0	0	1	1	3	ETX	DC3
0	1	0	0	4	EOT	DC4
0	1	0	1	5	ENQ	NAK
0	1	1	0	6	ACK	SYN
0	1	1	1	7	BEL	ETB
1	0	0	0	8	APB (BS)	CAN
1	0	0	1	9	APF (HT)	SS2
1	0	1	0	10	APD (LF)	SDC
1	0	1	1	11	APU (VT)	ESC
1	1	0	0	12	CS (FF)	APS
1	1	0	1	13	APR (CR)	SS3
1	1	1	0	14	SO	APH
1	1	1	1	15	SI	NSR

Figure 64

Jeu de commande C0

**6.1.2.4 INITIALISATION POSITION ACTIVE (APS).** Ce caractère (1/12) sert à déplacer le curseur sans réinitialiser les paramètres ni les attributs.

Les deux multiplets qui suivent immédiatement une commande APS doivent tous deux provenir des colonnes 2 à 7 ou 10 à 15 du tableau effectif. Ils représentent effectivement l'adresse de rangée et l'adresse de colonne reliées au déplacement du curseur. L'adresse de rangée est obtenue du premier multiplet qui suit une commande APS à partir du nombre entier binaire comprenant les bits b7 à b1, b7 étant le bit de poids fort, dont on soustrait 32 après avoir masqué b8. De la même façon, l'adresse de colonne est obtenue à partir du deuxième multiplet qui suit une commande APS, à partir du nombre entier binaire comprenant les bits b7 à b1, b7 étant le bit de poids fort, dont on soustrait 32. Cette méthode donne une plage d'adresses de 0 à 95 inclusivement pour les adresses de rangée et de colonne. Par exemple, la combinaison de bits 3/6 donne le nombre entier binaire 54, dont on soustrait 32 pour obtenir l'adresse 22. Si l'un des caractères qui suivent le caractère APS est une commande C0 ou C1, le caractère APS reste sans effet et la commande C0 ou C1 est exécutée.

Les rangées et les colonnes sont numérotées à partir de la rangée 0, colonne 0, position du caractère inférieur gauche de la plage d'affichage et se réfèrent au format d'écran nominal établi par les dimensions du champ de caractère actuel (avec l'espacement par défaut entre les caractères et entre les rangées). Le curseur est déplacé en supposant une rotation de caractère zéro pour établir l'origine du champ de caractère. Une fois cette origine établie, le champ de caractère et le curseur subissent une rotation, le cas échéant.

**6.1.2.5 POSITION ACTIVE VERS LE HAUT (APU).** Ce caractère (0/11) (tabulation verticale) sert à déplacer le curseur d'une distance égale à l'espace entre les rangées perpendiculairement au trajet de caractère, dans un sens perpendiculaire au trajet (90°). Lorsque le déplacement amènerait une partie quelconque du champ de caractère correspondant au-delà de la plage d'affichage (ou au-delà du champ actif si le champ de caractère se trouvait entièrement dans les limites du champs actif immédiatement avant le déplacement), des opérations spéciales sont exécutées en fonction du mode de défilement (avec ou sans défilement) (voir 6.2.7.13 et 6.2.7.14).

**6.1.2.6 EFFACEMENT D'ÉCRAN (CS).** Ce caractère (0/12) sert à positionner le curseur à la position du caractère supérieur gauche de la plage d'affichage, la partie supérieure du champ de caractère coïncidant avec la limite supérieure de la plage d'affichage. Dans les modes de couleur 0 et 1, cette commande efface la plage d'affichage et la fait passer au noir nominal. En mode de couleur 2, elle efface la plage d'affichage et la fait passer à la couleur de fond.

**6.1.2.7 RETOUR DE POSITION ACTIVE (APR).** Ce caractère (0/13) (retour de chariot) sert à ramener le curseur à la position du premier caractère de la plage d'affichage (ou dans les limites du champ actif, si le champ de caractère correspondant au curseur se trouvait entièrement dans les limites du champ actif avant le déplacement) dans le sens du trajet de caractère.

**6.1.2.8 POSITION ACTIVE INITIALE (APH).** Ce caractère 1/14 sert à ramener le curseur à la position du caractère supérieur gauche de la plage d'affichage, la partie supérieure du champ de caractère coïncidant avec la limite supérieure de la plage d'affichage.



### **6.1.3 Caractères de commande d'extension de code**

**6.1.3.1 HORS CODE (SO).** Ce caractère (0/14) sert à appeler le jeu G1 dans le tableau effectif (voir 4.3.2 et 4.3.3).

**6.1.3.2 EN CODE (SI).** Ce caractère (0/15) sert à appeler le jeu G0 dans le tableau effectif (voir 4.3.2 et 4.3.3).

**6.1.3.3 INVERSION UNIQUE DEUX (SS2).** Ce caractère (1/9) sert à appeler le jeu G2 dans le tableau effectif, sans verrouillage (voir 4.3.2 et 4.3.3).

**6.1.3.4 INVERSION UNIQUE TROIS (SS3).** Ce caractère (1/13) sert à appeler le jeu G3 dans le tableau effectif, sans verrouillage (voir 4.3.2 et 4.3.3).

**6.1.3.5 ÉCHAPPEMENT (ESC).** Ce caractère (1/11) sert à l'extension de code (voir 4.3.2 et 4.3.3).

**6.1.4 Caractères de commande de transmission.** Les caractères de commande de transmission, SOH (0/1), STX (0/2), ETX (0/3), EOT (0/4), ENQ (0/5), ACK (0/6), DLE (1/0), NAK (1/5), SYN (1/6) et ETB (1/7) restent sans effet sur la couche présentation; ils sont réservés à d'autres couches de protocole. Ils peuvent être emboîtés dans toute séquence de la couche présentation sans influencer sur cette séquence.

**6.1.5 Caractères de commande de périphérique.** Les caractères de commande de périphérique, DC1 (1/1), DC2 (1/2), DC3 (1/3) et DC4 (1/4) restent sans effet sur la couche présentation, ils sont réservés à d'autres couches de protocole. Ils peuvent être emboîtés dans toute séquence de la couche présentation sans influencer sur cette séquence.

### **6.1.6 Caractères de commande divers**

**6.1.6.1 NUL (NUL).** Ce caractère (0/0) reste sans effet sur la couche présentation; il est réservé à d'autres couches de protocole. Il peut être emboîté dans toute séquence de la couche présentation sans influencer sur cette séquence.

**6.1.6.2 SONNERIE (BEL).** Ce caractère (0/7) sert à déclencher momentanément une sonnerie ou à déclencher toute autre indication transitoire.

**6.1.6.3 ANNULATION (CAN).** Ce caractère (1/8) sert à mettre fin au traitement de toutes les macro-instructions en cours d'exécution. L'exécution reprend au caractère de la couche présentation qui suit l'annulation. L'incidence du code CAN est immédiate, c'est-à-dire que le code n'est pas placé à la fin d'une file existante de codes non traités de la couche présentation. La fonction du caractère CAN n'est garantie que si elle est garantie par les couches inférieures.

**6.1.6.4 CARACTÈRE SÉPARATEUR DE SERVICE (SDC).** Ce caractère (1/10) doit être exécuté comme une opération vide au niveau de la couche présentation, toute autre utilisation étant liée à la matérialisation (voir l'annexe D).

**6.1.6.5 INITIALISATION NON SÉLECTIVE (NSR).** Ce caractère (1/15) sert à deux fonctions: il effectue l'initialisation non sélective du processus de présentation de la façon définie ci-dessous et il peut être utilisé comme solution de rechange pour positionner le curseur. La réception du caractère NSR provoque les événements ci-dessous.

(1) Les jeux G0, G1, G2, G3, C0 et C1 sont désignés dans l'état par défaut et le tableau de code effectif est appelé dans l'état par défaut, comme le décrit l'alinéa 4.3.

(2) Les paramètres DOMAINE prennent les valeurs par défaut, comme le décrit l'alinéa 5.3.2.2.

(3) Les paramètres de texte (provenant du code d'opération TEXTE, du jeu C1 et du champ actif) prennent les valeurs par défaut, comme le décrit l'alinéa 5.3.2.9.3.

(4) Les paramètres TEXTURE prennent les valeurs par défaut, comme le décrit l'alinéa 5.3.2.4. Les masques programmables ne sont pas effacés.

(5) Le mode de couleur passe au mode 0 et la couleur de dessin, au blanc nominal. La carte des couleurs reste inchangée.

(6) Si les deux multiplats qui suivent immédiatement le caractère NSR proviennent des colonnes 4 à 7 du tableau effectif, le curseur est positionné. Ces deux multiplats représentent, sous forme binaire (c'est-à-dire le chiffre binaire comprenant les bits b6 à b1, b6 étant le bit de poids fort), respectivement l'adresse de la rangée et de la colonne auxquelles le curseur doit être déplacé. Les rangées et les colonnes sont numérotées à partir de la rangée 0, colonne 0, position du caractère supérieur gauche de la plage d'affichage, et se réfèrent au format d'écran nominal établi par les dimensions de caractère par défaut. La partie supérieure du champ de caractère de la rangée 0 coïncide avec la limite supérieure de la plage d'affichage. Si l'un ou l'autre des deux multiplats qui suivent le caractère NSR ne provient pas des colonnes 4 à 7 (ou des colonnes 12 à 15) du tableau effectif, la fonction d'initialisation non sélective du caractère NSR est exécutée et le curseur n'est pas repositionné. Si les deux multiplats proviennent des colonnes 2 et 3 (ou des colonnes 10 et 11), ils restent sans effet. Si l'un ou l'autre des deux multiplats est un caractère de commande C0 ou C1 (colonnes 0,1 ou 8,9), il met fin à la séquence NSR et est exécuté.

					$b_8$	1 1
					$b_7$	0 0
					$b_6$	0 0
					$b_5$	0 1
						8 9
					$b_7$	1 1
					$b_6$	0 0
					$b_5$	0 1
						4 5
						A B
$b_4$	$b_3$	$b_2$	$b_1$	RANGÉE	COLONNE	
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		

DÉF MACRO	AVEC PROT.
DÉFP MACRO	EDC <sub>1</sub>
DÉPT MACRO	EDC <sub>2</sub>
DÉF JCDR	EDC <sub>3</sub>
DÉF TEXTURE	EDC <sub>4</sub>
FIN	AVEC RENOU. DE MOT
RÉPÉTITION	SANS RENOU. DE MOT
RÉPÉT. JUS. EOL	AVEC DÉFILE.
VIDÉO INVER.	SANS DÉFILE.
VIDÉO NORM.	DÉBUT SOULI.
TEXTE PETIT	FIN SOULI.
TEXTE MOYEN	CURSEUR INTERM.
TEXTE NORMAL	CURSEUR STABLE
TEXTE HAUTEUR	SUPPR. CURSEUR
DÉBUT CLIG.	FIN CLIG.
DOUBLE TAILLE	SANS PROT.

#### Définition des colonnes A et B

(1) Si une fonction de commande C1 est représentée par une séquence d'échappement à deux caractères (dans un code à 7 éléments), le tableau spécifie la combinaison de bits du caractère final en supposant: A = 4 et B = 5.

(2) Si une fonction de commande C1 est représentée par une seule combinaison à 8 éléments, le tableau spécifie cette combinaison en supposant: A = 8 et B = 9.

Figure 65  
Jeu de commande C1

## 6.2 Jeu de commande C1

6.2.1 Le jeu de commande C1 (voir figure 65) permet de commander le format de texte et de produire des macro-instructions, des caractères JCDR, des masques de texture programmables et des champs sans protection. Les détails sont donnés aux alinéas 6.2.2 à 6.2.6.

### 6.2.2 Généralités sur les macro-instructions

6.2.2.1 **DEF MACRO.** Cette commande C1 sert à définir une macro-instruction. Les bits b7 à b1 du caractère qui suit cette commande doivent se situer dans la plage 2/0 à 7/15; ils constituent la combinaison de bits représentant la macro-instruction à définir. Tous les caractères subséquents sont mémorisés (mais non traités) dans l'appareil récepteur, où ils sont associés à macro-instruction spécifiée. Le macro-code prend fin dès la réception de l'une des commandes C1 suivantes: DEF MACRO, DEFP MACRO, DEFT MACRO, DEF JCDR, DEF TEXTURE ou FIN. Ni le caractère de commande de terminaison ni le caractère ESC précédent, dans le cas d'un codage à 7 éléments, ne sont mémorisés en tant que partie intégrante du macro-code. Si le caractère qui suit la commande DEF MACRO ne se situe pas dans cette plage, toute la commande (c'est-à-dire la commande C1 et le caractère hors plage) est erronée et est exécutée comme une opération vide.

Dans un macro-code, toute référence à une macro-instruction entraîne la mémorisation de cette référence seulement, et non pas son remplacement par le macro-code associé. Le macro-code remplace tout macro-code déjà associé à même macro-instruction. Un macro-code peut comporter plus ou moins de caractères que le macro-code précédent qu'il remplace. Un macro-code nul, c'est-à-dire qui ne comporte aucun caractère entre la macro-instruction et le caractère C1 de terminaison (ou le caractère ESC qui le précède dans le cas d'un codage à 7 éléments), entraîne la suppression de la macro-instruction. Le macro-code n'est pas liée au fait que le jeu de macro-instructions soit appelé ou non (bien qu'il doive évidemment être appelé pour que le macro-code soit effectivement exécuté).

Toutes les macro-instructions peuvent être supprimées simultanément à l'aide de la commande INITIALISATION.

6.2.2.2 **DEFP MACRO.** La fonction de ce caractère de commande est identique à celle de la commande DEF MACRO, sauf que les caractères d'entrée qui constituent le macro-code sont traités et mémorisés simultanément. Une macro-instruction est considérée comme étant non définie durant la définition jusqu'à ce que la définition prenne fin. Par conséquent, si une commande DEFP MACRO contient un renvoi à elle-même, ou si elle appelle une autre macro-instruction qui rappelle la macro-instruction en cours de définition, le renvoi à la macro-instruction en cours de définition est exécuté comme une opération vide.

**6.2.2.3 DÉFT MACRO.** Le caractère de commande DÉFT MACRO permet de définir une macro-instruction de transmission. Lorsqu'elle sont appelée, la macro-instruction de transmission n'est pas exécutée. Le macro-code associé est plutôt retourné en entier à l'ordinateur central ou à un appareil d'application locale. La touche de caractère peut devenir une touche à fonction programmable sur les claviers capables de commander l'exécution des macro-instructions par la frappe d'une ou plusieurs touches.

Les macro-instructions de transmission sont définies et supprimées de la même façon que les macro-instructions normales, et ils partagent les 96 mêmes macro-instructions.

**6.2.3 DÉF JCDR.** La commande DÉF JCDR permet d'amorcer le téléchargement de l'un des 96 caractères JCDR autorisés. Les bits b7 à b1 du caractère qui suit cette commande doivent se situer dans la plage 2/0 à 7/15; ils constituent la combinaison de bits qui représente le caractère JCDR défini. Vient ensuite tout code de données valides du code de la couche présentation. L'opération de téléchargement JCDR prend fin à la réception d'une commande FIN, DÉF MACRO, DÉFP MACRO, DÉFT MACRO, DÉF TEXTURE ou d'une autre commande DÉF JCDR.

Lorsqu'une autre commande DÉF JCDR met fin à l'opération de téléchargement JCDR actuelle, le caractère suivant du jeu G JCDR (dans la séquence circulaire 2/0, 2/1, ... 2/15, 3/0, 3/1 ... 7/14, 7/15, 2/0, 2/1...) est défini par le code de la couche présentation qui suit immédiatement la nouvelle commande DÉF JCDR. (C'est la seule occasion où le caractère DÉF JCDR n'est pas suivi du caractère JCDR à définir.)

Si les bits b7 à b1 du caractère qui suit la commande DÉF JCDR ne se trouvent pas dans la plage 2/0 à 7/15 et que la commande DÉF JCDR ne met pas fin à une commande DÉF JCDR précédente, toute la commande (c'est-à-dire la commande C1 et le caractère hors plage) est erronée et est exécutée comme une opération vide. Lorsqu'une définition JCDR prend fin immédiatement sans intervention d'un code de la couche présentation, l'espace affecté à ce caractère dans la mémoire tampon est libéré.

La définition d'un jeu JCDR n'est pas liée au fait que le jeu soit appelé ou non (bien qu'il doive évidemment être appelé dans le tableau effectif pour que le caractère soit effectivement affiché).

La chaîne de codes de la couche présentation définissant le caractère JCDR doit être traité dès sa réception. Elle est exécutée sur l'écran unitaire mais n'apparaît pas sur l'écran d'affichage physique. Cette exécution a pour effet de modifier un tampon de mémoire distinct, utilisé pour tout affichage subséquent du caractère JCDR. L'effet sur l'écran d'affichage physique de la redéfinition ou de la suppression d'un caractère JCDR n'est pas défini.

Tous les codes de la couche présentation maintiennent leur effet normal sur l'état de l'appareil récepteur, sauf que toutes les opérations de dessin modifient le tampon de mémoire JCDR plutôt que la plage d'affichage. Par exemple, si la carte des couleurs est changée, toute partie de l'écran d'affichage utilisant la nouvelle carte changera de couleur. Tout changement de l'état de l'appareil récepteur, par exemple des dimensions du champ de caractère et des jeux G effectifs, doit se maintenir après la définition JCDR.

Le rapport de forme du tampon de mémoire doit être identique à celui des dimensions du champ de caractère lorsque le caractère DÉF JCDR est reçu. Le coin inférieur gauche du tampon doit coïncider avec le coin inférieur gauche de l'écran unitaire. La dimension la plus grande du tampon (dx ou dy) doit coïncider entièrement avec l'axe correspondant de l'écran unitaire. Par exemple, un champ de caractère de 6/256 (de l'écran unitaire) de largeur et de 10/256 de hauteur a pour effet que la forme du caractère JCDR est définie seulement par un code écrivant dans la partie de l'écran unitaire située dans la plage 0.0 à 0.6 sur l'axe des X et 0.0 (inclusivement) à 1.0 (exclusivement) sur l'axe des Y. Si les dimensions du champ de caractère sont changées dans la définition JCDR, ce changement reste sans effet sur le rapport de forme et la résolution du tampon de mémoire, mais influe sur l'exécution des caractères subséquents dans l'écran unitaire.

Le tampon de mémoire de chaque caractère JCDR est divisé en éléments. Chaque élément peut se trouver dans l'état faux ou dans l'état vrai. Au commencement de chaque définition JCDR, tous les éléments du tampon de mémoire sont placés dans l'état faux par l'appareil récepteur. L'état de chaque élément du tampon de mémoire doit être défini par le code de la couche présentation écrivant dans la partie de l'écran unitaire qui coïncide avec l'élément durant la définition JCDR. Chaque élément touché doit être placé dans l'état vrai, à moins d'être maintenu à l'état faux par l'écriture du noir nominal. La résolution effective du tampon de mémoire est liée à la matérialisation.

Une fois la séquence de téléchargement terminée, l'appareil récepteur revient à son état normal, qui consiste à effectuer la transposition de l'écran unitaire à l'écran d'affichage tandis que la position de dessin est remise à (0,0).

Tout le jeu de caractères JCDR peut être supprimé à l'aide de la commande INITIALISATION. Lorsqu'une commande INITIALISATION de suppression JCDR est reçue durant une opération de téléchargement, la définition en cours du profil de caractère prend fin, mais l'opération de téléchargement se poursuit.

L'affichage d'un caractère JCDR a pour effet de mettre à l'échelle le contenu du tampon de mémoire de caractère pour qu'il s'ajuste au champ de caractère actuel. Les éléments à l'état vrai doivent être affichés dans la couleur de dessin au moment de l'affichage, et non pas dans la couleur de dessin au moment de la définition. En mode de couleur 2, les éléments à l'état faux doivent être affichés dans la couleur de fond. L'affichage des caractères JCDR doit porter tous les attributs de TEXTE. À noter que si les dimensions du champ de caractère actuel ne sont pas identiques aux dimensions du champ de caractère au commencement de la définition JCDR, ni un multiple entier de ces dimensions, une certaine déformation peut se produire en raison de la conversion d'échelle et de l'interpolation. À noter également qu'un tampon de mémoire correspondant à la description ci-dessus ne constitue pas la seule façon d'obtenir les effets définis.

**6.2.4 DÉF TEXTURE.** Cette commande permet de définir l'un des quatre masques de texture programmables, de la façon décrite à l'alinéa 5.3.2.4.

Les bits b7 à b1 du caractère qui suit cette commande doivent se trouver parmi l'une des combinaisons de bits (4/1), (4/2), (4/3), 4/4) entraînant respectivement la définition du masque A, B, C ou D. Tout motif existant associé au masque spécifié est supprimé. Le masque est effacé en mettant fin à la commande à ce moment. Si le code de la couche présentation vient ensuite, il décrit le masque de texture de la même façon que les caractères JCDR, sauf que les dimensions du masque de texture sont utilisés plutôt que les dimensions du champ de caractère. La commande DÉF TEXTURE prend fin à la réception d'une commande FIN, DÉF MACRO, DÉFP MACRO, DÉFT MACRO, DÉF JCDR ou d'une autre commande DÉF TEXTURE. Si les bits b7 à b1 du caractère qui suit la commande DÉF TEXTURE ne se trouvent pas dans la plage 4/1 à 4/4, toute la commande (c'est-à-dire la commande C1 et le caractère hors plage) est erronée et est exécutée comme une opération vide. À la fin de la commande DÉF TEXTURE, l'appareil récepteur revient à son état normal, qui consiste à effectuer la transposition de l'écran unitaire à l'écran d'affichage tandis que la position de dessin est remise à (0,0).

**Note:** La commande INCRÉMENTATION POINT peut convertir l'échelle du champ actif avant l'exécution (voir 5.3.3.6.3), les dimensions de la plage actuelle définie devenant inférieures à ce qui est attendu.

**6.2.5 FIN.** Cette commande met un terme à toute opération actuelle DÉF MACRO, DÉFP MACRO, DÉFT MACRO, DÉF JCDR ou DÉF TEXTURE. Elle sert également à transmettre des données dans un champ sans protection (voir 6.2.6).

**6.2.6 AVEC PROTECTION/SANS PROTECTION.** Les champs sans protection sont des zones rectangulaires de l'affichage dans lesquelles l'utilisateur peut introduire ou éditer des données en vue d'une éventuelle transmission. La méthode d'introduction et d'édition de données dans ces champs est liée à la matérialisation.

Dans la condition par défaut, tout l'écran d'affichage fait l'objet d'une protection, c'est-à-dire que l'utilisateur ne peut pas introduire ni modifier des données sur l'écran unitaire. Toutefois, lorsque la commande SANS PROTECTION est produite, le champ actif (défini par la commande CHAMP décrite à l'alinéa 5.3.3.6.2) passe au mode sans protection et l'utilisateur peut subséquemment introduire ou éditer localement des données à l'intérieur de ce champ. Il est possible de définir n'importe quel nombre de champs sans protection (compte tenu des limites de mémoire liées à la matérialisation), en définissant un champ actif par l'intermédiaire de la commande CHAMP, suivie de la commande SANS PROTECTION. Si une commande SANS PROTECTION est appliquée à un champ qui se trouve déjà partiellement ou entièrement dans un champ sans protection, le champ sans protection passe au mode AVEC PROTECTION (sans incidence sur le contenu affiché) avant que le nouveau champ passe au mode sans protection. Il s'agit par là d'empêcher le chevauchement des champs sans protection.

Les données reçues et affichées dans un champ sans protection après que ce champ est passé au mode sans protection, lorsque ce champ sans protection coïncide avec le champ actif, peuvent également être éditées subséquentement par l'utilisateur. Lorsque l'utilisateur amorce une transmission, l'information des champs sans protection est transmise d'après la présente norme. La commande CHAMP renfermant les coordonnées d'origine ainsi que les dimensions du champ sans protection est transmise en premier lieu, puis elle est suivie du contenu du champ transféré et finalement de la commande FIN. En présence de plusieurs champs sans protection, l'ordre de transmission va de la partie supérieure à la partie inférieure de l'écran unitaire, l'origine des champs constituant la référence. Lorsque deux ou plusieurs origines ont la même coordonnée Y, l'ordre de transmission commence par la gauche. La présentation de transmission, avec son état associé, doit rester distincte de la présentation de réception.

Les champs sans protection peuvent être protégés à nouveau au moyen de la commande AVEC PROTECTION. Cette commande permet de protéger tous les champs sans protection dont une partie quelconque se trouve dans le champ actif à protéger. La commande INITIALISATION peut servir à protéger tous les champs sans protection actuellement définis.

Le recours à des champs sans protection n'empêche pas l'utilisateur de faire appel à d'autres modes d'entrée indépendants de l'état des champs sans protection et sans incidence sur l'état de ces champs. Les modes d'entrée de rechange sont liés à la matérialisation et peuvent temporairement utiliser la plage d'affichage.

#### **6.2.7 Commandes de texte**

**6.2.7.1** Ce jeu de 20 commandes permet de matérialiser des fonctions de texte simples, dont certaines peuvent également être exécutées par l'intermédiaire des commandes TEXTE ou CLIGNOTEMENT.

**6.2.7.2 RÉPÉTITION.** Cette commande permet de répéter un nombre déterminé de fois le dernier multipllet reçu s'il s'agit d'un multipllet ESPACE ou de tout caractère d'espacement provenant des jeux primaire, supplémentaire, JCQR ou mosaïque, sinon la commande est erronée et doit être exécutée comme une opération vide. Les bits b6 à b1 du caractère qui suit le caractère RÉPÉTITION doivent être interprétés comme le compte de répétition... Les bits b7 à b1 de ce caractère de compte de répétition doivent se trouver dans la plage 4/0 à 7/15, sinon la commande est erronée et doit être exécutée comme une opération vide, le caractère de compte étant exécuté comme un caractère provenant des colonnes 0 à 3 ou 8 à 11.

**6.2.7.3 RÉPÉTITION JUSQU'À EOL.** Lorsque cette commande est produite, le dernier multipllet est répété jusqu'à ce que soit atteinte la dernière position de caractère du trajet actuel s'il s'agit d'un multipllet ESPACE ou de tout caractère d'espacement provenant des jeux primaire, supplémentaire, JCQR ou mosaïque, sinon la commande est erronée et doit être exécutée comme une opération vide. Si le champ de caractère correspondant au curseur de texte se trouve entièrement dans le champ actif lorsque cette commande se présente, les caractères se répètent alors seulement jusqu'à la dernière position de caractère du trajet actuel à l'intérieur du champ actif.



**6.2.7.4 VIDÉO INVERSÉE.** Cette commande fait passer l'appareil récepteur en mode de vidéo inversée, dans lequel tout caractère alphanumérique de texte, mosaïque et JCDR subséquentment reçu est dessiné de façon que les pixels entourant la forme du caractère dans le champ de caractère prennent la couleur de dessin. Les pixels de la forme de caractère ne sont pas dessinés, sauf en mode de couleur 2, dans lequel ils prennent la couleur de fond.

**6.2.7.5 VIDÉO NORMALE.** Cette commande fait sortir l'appareil récepteur du mode de vidéo inversée. Il s'agit de l'état par défaut.

**6.2.7.6 TEXTE PETIT.** Cette commande fait passer les dimensions du champ de caractère à  $dx = 1/80$  et  $dy = 5/128$ , compte tenu de la résolution physique.

**6.2.7.7 TEXTE MOYEN.** Cette commande fait passer les dimensions du champ de caractère à  $dx = 1/32$  et  $dy = 3/64$ , compte tenu de la résolution physique.

**6.2.7.8 TEXTE NORMAL.** Cette commande fait passer les dimensions du champ de caractère à leur valeur par défaut, soit  $dx = 1/40$  et  $dy = 5/128$ , compte tenu de la résolution physique.

**6.2.7.9 DOUBLE HAUTEUR.** Cette commande fait passer les dimensions du champ de caractère à  $dx = 1/40$  et  $dy = 5/64$ , compte tenu de la résolution physique.

**6.2.7.10 DOUBLE TAILLE.** Cette commande fait passer les dimensions du champ de caractère à  $dx = 1/20$  et  $dy = 5/64$ , compte tenu de la résolution physique.

À noter que les commandes TEXTE PETIT, TEXTE MOYEN, TEXTE NORMAL, DOUBLE HAUTEUR et DOUBLE TAILLE influent seulement sur le signe et la grandeur des dimensions du champ de caractère. Elles n'influencent pas sur la rotation, le trajet de caractère, l'espacement des caractères et l'espacement des rangées, la vidéo inversée, le défilement, le renouement de mot, le soulignement, l'état du curseur, le style du curseur ni sur les attributs de déplacement.

**6.2.7.11 AVEC RENOUEMENT DE MOT.** Cette commande fait entrer l'appareil récepteur en mode de renouement de mot. Dans ce mode, le texte alphanumérique subséquentment reçu est mis en tampon par mots. Un mot est affiché sur la ligne actuelle de l'écran seulement lorsque tout le mot en tampon peut entrer dans l'espace qui reste sur la ligne actuelle de l'écran unitaire (ou du champ actif, si le champ de caractère correspondant au curseur de texte se trouve entièrement dans le champ actif). Lorsque le mot n'entre pas dans l'espace qui reste sur la ligne actuelle, une séquence APR et APD est exécutée automatiquement et le mot est affiché. Le caractère ESPACE doit être omis si le dernier mot de la ligne se termine par un ESPACE qui n'entre pas sur cette ligne. Aux fins du présent alinéa, un mot est défini comme étant une accumulation de caractères entre des caractères ESPACE.

Les mots entièrement formés de caractères alphabétiques (voir tableau 18) et d'un ou plusieurs caractères spéciaux de terminaison (! " \$ % ( ) [ ] ^ \* + - / , . : ; = ? \_ ~ ) emboîtés (c'est-à-dire ni au début ni à la fin du mot) peuvent être séparés entre le caractère spécial de terminaison et le caractère suivant, de façon à placer la plus grande partie possible du mot sur la ligne actuelle. Tous les autres mots doivent être placés sur une seule ligne.

Un mot peut également se terminer par un caractère du jeu mosaïque, un code IDI, tout caractère du jeu C défini au niveau de la couche présentation sauf SO, SI, SS2, et SS3, ou tout caractère qui rend la longueur du mot égale à la longueur maximale de la ligne.

**6.2.7.12 SANS RENOUEMENT DE MOT.** Cette commande fait sortir l'appareil récepteur du mode renouement de mot. Dans ce mode (mode par défaut), tout le texte est coupé aux limites des caractères à chaque exécution automatique des codes APR et APD.

**6.2.7.13 AVEC DÉFILEMENT.** Dans ce mode, une commande APB ou APU ou une séquence automatique APR APD qui ferait sortir une partie quelconque du curseur de la plage d'affichage (ou du champ actif, si le champ de caractère correspondant au curseur se trouve entièrement dans les limites du champ actif) fait défiler tout l'affichage à l'intérieur de la plage ou du champ. Le défilement se fait pixel par pixel, perpendiculairement au trajet de caractère, sur une distance suffisante pour amener la prochaine position de caractère prévue jusqu'à l'intérieur de la plage ou du champ. La couleur des pixels de fond amenés dans la plage ou le champ est le noir nominal dans les modes de couleur 0 et 1 et la couleur de fond, en mode de couleur 2.

Le fait que les données expulsées de la plage d'affichage ou du champ actif soient mises en tampon ou non est lié à la matérialisation.

**6.2.7.14 SANS DÉFILEMENT.** Dans ce mode (mode par défaut), une commande APD ou APU ou une séquence automatique APR APD qui ferait sortir une partie quelconque du curseur de la plage d'affichage (ou du champ actif, si le champ de caractère correspondant au curseur se trouve entièrement dans les limites du champ actif) repositionne le curseur du côté opposé de la plage ou du champ, de façon que le champ de caractère ainsi défini se trouve entièrement à l'intérieur de la plage ou du champ.

**6.2.7.15 DÉBUT SOULIGNEMENT.** Cette commande fait entrer l'appareil récepteur en mode de soulignement. Dans ce mode, une ligne est ajoutée à toutes les configurations des caractères primaires, supplémentaires, JCDR et ESPACE. La ligne apparaît dans le champ de caractère à partir de l'origine du champ et s'étend sur toute la largeur du champ (dimension dx) mais pas sur la largeur de l'espace entre les champs de caractère lorsque l'espacement entre les caractères est supérieur à l'unité. Son épaisseur est déterminée par la dimension verticale (dy) du pel logique. Dans ce mode, tous les caractères du jeu mosaïque sont affichés sous des formes mosaïques distinctes (voir 5.4). Les caractères mosaïques ne sont pas soulignés.

**6.2.7.16 FIN SOULIGNEMENT.** Cette commande fait sortir l'appareil récepteur du mode soulignement. Dans ce mode (mode par défaut), les caractères du jeu mosaïque sont affichés de façon contiguë, c'est-à-dire que les six éléments mosaïques composant chaque caractère s'étendent sur toute la plage normale.

**6.2.7.17 CURSEUR INTERMITTENT.** Cette commande affiche le curseur de façon intermittente, permet l'introduction de données par l'utilisateur (voir 6.2.6) et la mise en jeu par l'utilisateur de macro-instructions enchaînées (voir 5.5).

**6.2.7.18 CURSEUR STABLE.** Cette commande affiche le curseur de façon stable, permet l'introduction de données par l'utilisateur (voir 6.2.6) et la mise en jeu par l'utilisateur de macro-instructions (voir 5.5).

**6.2.7.19 SUPPRESSION CURSEUR.** Cette commande rend le curseur invisible (mode par défaut). À noter que le curseur remplit toujours ses fonctions et se déplace comme à l'ordinaire, mais qu'il n'est tout simplement pas visible dans ce mode. Le fait que la commande SUPPRESSION CURSEUR puisse ou non invalider ou mettre en tampon les données introduites par l'utilisateur, rester sans effet sur ces données (voir 6.2.6), invalider l'association des macro-instructions à un mécanisme d'entrée de l'utilisateur ou rester sans effet sur cette association (voir 5.5), est lié à la matérialisation.

## **6.2.8 COMMANDES DIVERSES**

**6.2.8.1 DÉBUT CLIGNOTEMENT.** Cette commande établit le processus de clignotement suivant: la couleur repos est la couleur de dessin; la couleur travail est le noir nominal dans les modes de couleur 0 et 1, ou la couleur de fond en mode de couleur 2; les intervalles travail et repos sont liés à la matérialisation et le retard de phase est 0.

En mode de couleur 0, la commande de DÉBUT CLIGNOTEMENT C1 établit un processus automatique qui définit implicitement une couleur de clignotement. À noter que les objets dessinés dans la couleur de dessin précédente restent sans clignotement, mais que le clignotement s'applique aux objets subséquentement dessinés dans la nouvelle couleur de dessin. S'il y a changement de la couleur de dessin, l'ancienne couleur reste avec clignotement et la nouvelle couleur se produit sans clignotement.

**6.2.8.2 FIN CLIGNOTEMENT.** Cette commande (condition par défaut) met fin à tout processus actuel de clignotement dans lequel la couleur de dessin sert de couleur repos.

**6.2.8.3 COMMANDES DE PÉRIPHÉRIQUE AVEC EXTENSION (EDC1, EDC2, EDC3 ET EDC4).** La signification précise des commandes EDC1, EDC2, EDC3 et EDC4 sera normalisée plus tard; ces commandes sont exécutées comme des opérations vides.

## 7. Répertoire des caractères graphiques

7.1 Les tableaux 18 à 26 présentent le répertoire des caractères graphiques et la représentation codée de ces caractères; le tableau 27 présente les mêmes caractères utilisés en combinaison avec le caractère ESPACE. Les caractères sans espacement du tableau 27 peuvent être utilisés en combinaison avec tout caractère graphique du répertoire, y compris un caractère accentué. Toutefois, de telles combinaisons ou d'autres combinaisons (par exemple la commande APB ou d'autres commandes de positionnement) ne constituent pas des caractères du répertoire. Tout caractère du répertoire des caractères graphiques doit pouvoir être affiché dans tous les modes de couleur. Cette exigence peut nécessiter une attention spéciale en mode de couleur 2 et en mode de vidéo inversée. La présentation des caractères qui ne font pas partie intégrante du répertoire des caractères graphiques est liée à la matérialisation.

7.2 Sur les tableaux 18 à 27, la colonne de la représentation codée spécifie la représentation codée des caractères graphiques des codes à 7 éléments et à 8 éléments. Le symbole S indique que le multiplet qui suit immédiatement représente un caractère du jeu supplémentaire. Sur les tableaux, les représentations codées sans le symbole S désignent des caractères du jeu primaire. Lorsque la représentation codée est constituée de combinaisons de deux multiplets (exemple: a minuscule avec accent aigu: S 4/2 6/1), le multiplet de gauche doit précéder celui de droite durant l'échange des informations. Dans les codes à 7 éléments et à 8 éléments, les caractères du jeu primaire sont représentés par des multiplets dans la plage de 2/1 à 7/14. Dans les codes à 7 éléments, les caractères du jeu supplémentaire doivent être représentés par SS2 ou SS3 suivis de 2/1 à 7/14 lorsque le jeu supplémentaire est désigné respectivement comme jeu G2 ou G3. Dans les codes à 8 éléments, les caractères du jeu supplémentaire doivent être représentés par SS2 suivi de 2/1 à 7/14 lorsque le jeu supplémentaire est désigné comme jeu G2 ou par 10/1 à 15/14 lorsque le jeu supplémentaire est appelé dans GR. Le nombre de multiplets nécessaires au codage des caractères accentués dépend de la séquence d'appel nécessaire (le cas échéant) pour appeler les jeux supplémentaire et primaire. La séquence d'appel du jeu primaire peut se produire après le caractère supplémentaire dans le codage des caractères accentués. Le répertoire des caractères est basé sur les normes ISO 646-1982 et ISO 6937-1982, ainsi que sur les Avis V3(1972) et S.100-1980 du CCITT.

7.3 Sur les tableaux 18 à 27, les représentations codées désignent simplement le jeu primaire et le jeu supplémentaire sans référence aux jeux G faisant l'objet de la désignation et de l'appel. Dans la présente syntaxe de données, le jeu primaire peut être désigné et appelé comme jeu G0, G1, G2 ou G3, le jeu G0 étant le jeu par défaut, et le jeu supplémentaire peut être désigné et appelé comme jeu G0, G1, G2 ou G3, le jeu G2 étant le jeu par défaut.

Tableau 18  
Caractères de l'alphabet latin

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
a	a minuscule	6/1
A	A majuscule	4/1
á	a minuscule avec accent aigu	S 4/2 6/1
Á	A majuscule avec accent aigu	S 4/2 4/1
à	a minuscule avec accent grave	S 4/1 6/1
À	A majuscule avec accent grave	S 4/1 4/1
â	a minuscule avec accent circonflexe	S 4/3 6/1
Â	A majuscule avec accent circonflexe	S 4/3 4/1
ä	a minuscule avec tréma ou Umlaut	S 4/8 6/1
Ä	A majuscule avec tréma ou Umlaut	S 4/8 4/1
ã	a minuscule avec tilde	S 4/4 6/1
Ã	A majuscule avec tilde	S 4/4 4/1
ă	a minuscule avec signe de voyelle brève	S 4/6 6/1
Ă	A majuscule avec signe de voyelle brève	S 4/6 4/1
â	a minuscule avec rond	S 4/10 6/1
Â	A majuscule avec rond	S 4/10 4/1
ā	a minuscule avec signe de voyelle longue	S 4/5 6/1
Ā	A majuscule avec signe de voyelle longue	S 4/5 4/1
ą	a minuscule avec ogonek	S 4/14 6/1
Ą	A majuscule avec ogonek	S 4/14 4/1

æ	diphtongue æ minuscule	S 7/1
Æ	diphtongue Æ majuscule	S 6/1
b	b minuscule	6/2
B	B majuscule	4/2
c	c minuscule	6/3
C	C majuscule	4/3
ċ	c minuscule avec accent aigu	S 4/2 6/3
Ĉ	C majuscule avec accent aigu	S 4/2 4/3
ĉ	c minuscule avec accent circonflexe	S 4/3 6/3
Ĉ	C majuscule avec accent circonflexe	S 4/3 4/3
ċ̂	c minuscule avec accent circonflexe inversé	S 4/15 6/3
Ĉ̂	C majuscule avec accent circonflexe inversé	S 4/15 4/3
ċ	c minuscule avec point	S 4/7 6/3
Ĉ	C majuscule avec point	S 4/7 4/3
ç	c minuscule avec cédille	S 4/11 6/3
Ç	C majuscule avec cédille	S 4/11 4/3
d	d minuscule	6/4
D	D majuscule	4/4
ḏ ou ḏ̂	d minuscule avec accent circonflexe inversé ou apostrophe	S 4/15 6/4
Ḑ	D majuscule avec accent circonflexe inversé	S 4/15 4/4

đ	d minuscule barré	S 7/2
Ð	D majuscule barré, (équivalent islandais du th doux anglais)	S 6/2
ḍ	d minuscule barré (équivalent islandais du th doux anglais)	S 7/3
e	e minuscule	6/5
E	E majuscule	4/5
é	e minuscule avec accent aigu	S 4/2 6/5
É	E majuscule avec accent aigu	S 4/2 4/5
è	e minuscule avec accent grave	S 4/1 6/5
È	E majuscule avec accent grave	S 4/1 4/5
ê	e minuscule avec accent circonflexe	S 4/3 6/5
Ê	E majuscule avec accent circonflexe	S 4/3 4/5
ë	e minuscule avec tréma ou Umlaut	S 4/8 6/5
Ë	E majuscule avec tréma ou Umlaut	S 4/8 4/5
ẽ	e minuscule avec accent circonflexe inversé	S 4/15 6/5
Ê	E majuscule avec accent circonflexe inversé	S 4/15 4/5
ė	e minuscule avec point	S 4/7 6/5
Ė	E majuscule avec point	S 4/7 4/5
ē	e minuscule avec signe de voyelle longue	S 4/5 6/5
Ē	E majuscule avec signe de voyelle longue	S 4/5 5/5
ę	e minuscule avec ogonek	S 4/14 6/5
Ę	E majuscule avec ogonek	S 4/14 4/5

f	f minuscule	6/6
F	F majuscule	4/6
g	g minuscule	6/7
G	G majuscule	4/7
g	g minuscule avec accent aigu	S 4/2 6/7
g	g minuscule avec accent circonflexe	S 4/3 6/7
G	G majuscule avec accent circonflexe	S 4/3 4/7
g	g minuscule avec signe de voyelle brève	S 4/6 6/7
G	G majuscule avec signe de voyelle brève	S 4/6 4/7
g	g minuscule avec point	S 4/7 6/7
G	G majuscule avec point	S 4/7 4/7
G	G majuscule avec cédille	S 4/11 4/7
h	h minuscule	6/8
H	H majuscule	4/8
h	h minuscule avec accent circonflexe	S 4/3 6/8
H	H majuscule avec accent circonflexe	S 4/3 4/8
h	h minuscule barré	S 7/4
H	H majuscule barré	S 6/4
i	i minuscule	6/9
I	I majuscule	4/9
i	i minuscule avec accent aigu	S 4/2 6/9
I	I majuscule avec accent aigu	S 4/2 4/9



ï	i minuscule avec accent grave	S 4/1 6/9
Î	I majuscule avec accent grave	S 4/1 4/9
î	i minuscule avec accent circonflexe	S 4/3 6/9
Î	I majuscule avec accent circonflexe	S 4/3 4/9
ï	i minuscule avec tréma ou Umlaut	S 4/8 6/9
Ï	I majuscule avec tréma ou Umlaut	S 4/8 4/9
ĩ	i minuscule avec tilde	S 4/4 6/9
Ĩ	I majuscule avec tilde	S 4/4 4/9
İ	I majuscule avec point	S 4/7 4/9
ī	i minuscule avec signe de voyelle longue	S 4/5 6/9
Ī	I majuscule avec signe de voyelle longue	S 4/5 4/9
į	i minuscule avec ogonek	S 4/14 6/9
Į	I majuscule avec ogonek	S 4/14 4/9
ij	ligature ij minuscule	S 7/6
IJ	ligature IJ majuscule	S 6/6
ı	i minuscule sans point	S 7/5
j	j minuscule	6/10
J	J majuscule	4/10
ĵ	j minuscule avec accent circonflexe	S 4/3 6/10
Ĵ	J majuscule avec accent circonflexe	S 4/3 4/10
k	k minuscule	6/11
K	K majuscule	4/11

<b>k</b>	k minuscule avec cédille	S 4/11 6/11
<b>K</b>	K majuscule avec cédille	S 4/11 4/11
<b>ƙ</b>	k minuscule groenlandais	S 7/0
<b>l</b>	l minuscule	6/12
<b>L</b>	L majuscule	4/12
<b>ï</b>	l minuscule avec accent aigu	S 4/2 6/12
<b>Ĺ</b>	L majuscule avec accent aigu	S 4/2 4/12
<b>ı ou ĩ</b>	l minuscule avec accent circonflexe inversé ou apostrophe	S 4/15 6/12
<b>Ĺ ou Ľ</b>	L majuscule avec accent circonflexe inversé ou apostrophe	S 4/15 4/12
<b>ł</b>	l minuscule avec cédille	S 4/11 6/12
<b>Ł</b>	L majuscule avec cédille	S 4/11 4/12
<b>ł̸</b>	l minuscule barré en oblique	S 7/8
<b>Ł̸</b>	L majuscule barré en oblique	S 6/8
<b>ḷ</b>	l minuscule avec point médian	S 7/7
<b>Ł̣</b>	L majuscule avec point médian	S 6/7
<b>m</b>	m minuscule	6/13
<b>M</b>	M majuscule	4/13
<b>n</b>	n minuscule	6/14
<b>N</b>	N majuscule	4/14
<b>ñ</b>	n minuscule avec accent aigu	S 4/2 6/14
<b>Ñ</b>	N majuscule avec accent aigu	S 4/2 4/14

ñ	n minuscule avec tilde	S 4/4 6/14
Ñ	N majuscule avec tilde	S 4/4 4/14
ny	n minuscule avec accent circonflexe inversé	S 4/15 6/14
NY	N majuscule avec accent circonflexe inversé	S 4/15 4/14
ñ	n miniscule avec cédille	S 4/11 6/14
Ñ	N majuscule avec cédille	S 4/11 4/14
ŋ	n minuscule lapon	S 7/14
Ŋ	N majuscule lapon	S 6/14
n	n minuscule avec apostrophe	S 6/15
o	o minuscule	6/15
O	O majuscule	4/15
ó	o minuscule avec accent aigu	S 4/2 6/15
Ó	O majuscule avec accent aigu	S 4/2 4/15
ò	o minuscule avec accent grave	S 4/1 6/15
Ò	O majuscule avec accent grave	S 4/1 4/15
ô	o minuscule avec accent circonflexe	S 4/3 6/15
Ô	O majuscule avec accent circonflexe	S 4/3 4/15
ö	o minuscule avec tréma ou Umlaut	S 4/8 6/15
Ö	o majuscule avec tréma ou Umlaut	S 4/8 4/15
õ	o minuscule avec tilde	S 4/4 6/15
Õ	O majuscule avec tilde	S 4/4 4/15

ó	o minuscule avec double accent aigu	S 4/13 6/15
Ó	O majuscule avec double accent aigu	S 4/13 4/15
ō	o minuscule avec signe de voyelle longue	S 4/5 6/15
Ō	O majuscule avec signe de voyelle longue	S 4/5 4/15
œ	ligature œ minuscule	S 7/10
Œ	ligature Œ majuscule	S 6/10
ø	o minuscule barré en oblique	S 7/9
Ø	O majuscule barré en oblique	S 6/9
p	p minuscule	7/0
P	P majuscule	5/0
q	q minuscule	7/1
Q	Q majuscule	5/1
r	r minuscule	7/2
R	R majuscule	5/2
ó	r minuscule avec accent aigu	S 4/2 7/2
Ó	R majuscule avec accent aigu	S 4/2 5/2
ṛ	r minuscule avec accent circonflexe inversé	S 4/15 7/2
Ṛ	R majuscule avec accent circonflexe inversé	S 4/15 5/2
ṛ	r minuscule avec cédille	S 4/11 7/2
Ṛ	R majuscule avec cédille	S 4/11 5/2
s	s minuscule	7/3
S	S majuscule	7/3

ś	s minuscule avec accent aigu	S 4/2 7/3
Ś	S majuscule avec accent aigu	S 4/2 5/3
ș	s minuscule avec accent circonflexe	S 4/3 7/3
Ș	S majuscule avec accent circonflexe	S 4/3 5/3
ȝ	s minuscule avec accent circonflexe inversé	S 4/15 7/3
Ț	S majuscule avec accent circonflexe inversé	S 4/15 5/3
ș	s minuscule avec cédille	S 4/11 7/3
Ș	S majuscule avec cédille	S 4/11 5/3
ß	s dur minuscule de l'allemand	S 7/11
t	t minuscule	7/4
T	T majuscule	5/4
ȧ ou ȧ	t minuscule avec accent circonflexe inversé ou apostrophe	S 4/15 7/4
Ȧ	T majuscule avec accent circonflexe inversé	S 4/5 5/4
ț	t minuscule avec cédille	S 4/11 7/4
Ț	T majuscule avec cédille	S 4/11 5/4
ⵜ	t minuscule barré	S 7/13
Ț	T majuscule barré	S 6/13
þ	t minuscule, équivalent islandais du th dur anglais	S 7/12
Þ	t majuscule, équivalent islandais du th dur anglais	S 6/12
u	u minuscule	7/5
U	U majuscule	5/5

ú	u minuscule avec accent aigu	S 4/2 7/5
Ú	U majuscule avec accent aigu	S 4/2 5/5
ù	u minuscule avec accent grave	S 4/1 7/5
Ù	U majuscule avec accent grave	S 4/1 5/5
û	u minuscule avec accent circonflexe	S 4/3 7/5
Û	U majuscule avec accent circonflexe	S 4/3 5/5
ü	u minuscule avec tréma ou Umlaut	S 4/8 7/5
Ü	U majuscule avec tréma ou Umlaut	S 4/8 5/5
ũ	u minuscule avec tilde	S 4/4 7/5
Ũ	U majuscule avec tilde	S 4/4 5/5
ṽ	u minuscule avec signe de voyelle brève	S 4/6 7/5
Ṫ	U majuscule avec signe de voyelle brève	S 4/6 5/5
ű	u minuscule avec double accent aigu	S 4/13 7/5
Ű	U majuscule avec double accent aigu	S 4/13 5/5
ũ	u minuscule avec rond	S 4/10 7/5
Ũ	U majuscule avec rond	S 4/10 5/5
ū	u minuscule avec signe de voyelle longue	S 4/5 7/5
Ū	U majuscule avec signe de voyelle longue	S 4/5 5/5
ȳ	u minuscule avec ogonek	S 4/14 7/5
Ȳ	U majuscule avec ogonek	S 4/14 5/5
v	v minuscule	7/6
V	V majuscule	5/6

w	w minuscule	7/7
W	W majuscule	5/7
ŵ	w minuscule avec accent circonflexe	S 4/3 7/7
Ŵ	W majuscule avec accent circonflexe	S 4/3 5/7
x	x minuscule	7/8
X	X majuscule	5/8
y	y minuscule	7/9
Y	Y majuscule	5/9
ÿ	y minuscule avec accent aigu	S 4/2 7/9
Ÿ	Y majuscule avec accent aigu	S 4/2 5/9
ÿ̂	y minuscule avec accent circonflexe	S 4/3 7/9
Ÿ̂	Y majuscule avec accent circonflexe	S 4/3 5/9
ÿ̇	y minuscule avec tréma ou Umlaut	S 4/8 7/9
Ÿ̇	Y majuscule avec tréma ou Umlaut	S 4/8 5/9
z	z minuscule	7/10
Z	Z majuscule	5/10
ż	z minuscule avec accent aigu	S 4/2 7/10
Ż	Z majuscule avec accent aigu	S 4/2 5/10
ẑ	z minuscule avec accent circonflexe inversé	S 4/15 7/10
Ẑ	Z majuscule avec accent circonflexe inversé	S 4/15 5/10
ẓ	z minuscule avec point	S 4/7 7/10
Ẓ	Z majuscule avec point	S 4/7 5/10

Tableau 19  
Chiffres décimaux

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
1	chiffre 1	3/1
2	chiffre 2	3/2
3	chiffre 3	3/3
4	chiffre 4	3/4
5	chiffre 5	3/5
6	chiffre 6	3/6
7	chiffre 7	3/7
8	chiffre 8	3/8
9	chiffre 9	3/9
0	chiffre 0	3/0

Tableau 20  
Symboles monétaires

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
⌘	symbole monétaire international	S 2/8
£	symbole de la livre	S 2/3
\$	symbole du dollar	2/4; S 2/4
¢	symbole du cent	S 2/2
¥	symbole du yen	S 2/5



Tableau 21  
Signes de ponctuation

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
	espace	2/0
!	point d'exclamation	2/1
¡	point d'exclamation inversé	S 2/1
"	guillemet droit	2/2
'	apostrophe	2/7
(	parenthèse d'ouverture (gauche)	2/8
)	parenthèse de fermeture (droite)	2/9
,	virgule	2/12
—	tiret inférieur ou souligné	5/15
-	trait d'union (signe moins)	2/13
.	point (point décimal)	2/14
/	barre oblique (solidus)	2/15
:	deux points	3/10
;	point-virgule	3/11
?	point d'interrogation	3/15
¿	point d'interrogation inversé	S 3/15
«	guillemet anguleux d'ouverture (gauche)	S 2/11
»	guillemet anguleux de fermeture (droite)	S 3/11
‘	guillemet tronqué d'ouverture (gauche)	S 2/9
’	guillemet tronqué de fermeture (droite)	S 3/9
“	guillemet d'ouverture (gauche)	S 2/10
”	guillemet de fermeture (droite)	S 3/10

Tableau 22  
Signes arithmétiques

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
+	signe plus	2/11
$\pm$	signe plus ou moins	S 3/1
<	signe inférieur à	3/12
=	signe égal	3/13
>	signe supérieur à	3/14
-	signe divisé par	S 3/8
x	signe multiplié par	S 3/4

Tableau 23  
Indices inférieurs et supérieurs

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
1	indice supérieur 1	S 5/1
2	indice supérieur 2	S 3/2
3	indice supérieur 3	S 3/3

Tableau 24

Fractions

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
1/2	fraction un demi	S 3/13
1/4	fraction un quart	S 3/12
3/4	fraction trois quarts	S 3/14
1/8	fraction un huitième	S 5/12
3/8	fraction trois huitièmes	S 5/13
5/8	fraction cinq huitièmes	S 5/14
7/8	fraction sept huitièmes	S 5/15

Tableau 25  
Symboles divers

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
#	signe numéro	2/3; S 2/6
%	signe pour cent	2/5
&	perluète	2/6
*	astérisque	2/10
@	"a" commercial	4/0
[	crochet d'ouverture (gauche)	5/11
\	barre oblique inversée (solidus inversé)	5/12
]	crochet de fermeture (droite)	5/13
{	accolade d'ouverture (gauche)	7/11
	barre verticale	7/12
}	accolade de fermeture (droite)	7/13
μ	signe micro	S 3/5
Ω	signe ohm	S 6/0
°	signe degré	S 3/0
<u>o</u>	indicateur ordinal, masculin	S 6/11
<u>a</u>	indicateur ordinal, féminin	S 6/3
§	signe paragraphe	S 2/7
¶	signe alinéa à faire ou appel de note (en anglais)	S 3/6
.	point médian	S 3/7
←	flèche horizontale vers la gauche	S 2/12
→	flèche horizontale vers la droite	S 2/14
↑	flèche verticale vers le haut	S 2/13

	flèche verticale vers le bas	S 2/15
®	symbole marque déposée	S 5/2
©	symbole Copyright	S 5/3
TM.	symbole marque de fabrique	S 5/4
♪	symbole note de musique	S 5/5
`	accent grave	6/0
ˆ	accent circonflexe	5/14
~	tilde	7/14
▤	diagonale (voir note 1)	S 5/8
▥	diagonale inversée (voir note 2)	S 5/9
▦	diagonale remplie (voir note 3)	S 5/10
▧	diagonale inversée remplie (voir note 4)	S 5/11
⌘	croix (voir note 5)	S 6/5
▬	trait horizontal pleine longueur (voir note 6)	S 5/6
▮	trait vertical pleine longueur (voir note 7)	S 5/7
—	barre horizontale (voir note 8)	S 5/0

Notes:

- 1) La diagonale s'étend du coin inférieur gauche du champ de caractère jusqu'au coin supérieur droit du champ de caractère.
- 2) La diagonale inversée s'étend du coin inférieur droit du champ de caractère jusqu'au coin supérieur gauche du champ de caractère.
- 3) La diagonale remplie est un triangle rempli qui occupe la moitié inférieure droite du champ de caractère.
- 4) La diagonale remplie inversée est un triangle rempli qui occupe la moitié inférieure gauche du champ de caractère.

- 5) Le caractère croix est équivalent à superposer les deux caractères: trait horizontal pleine longueur et trait vertical pleine longueur.
- 6) Le trait horizontal pleine longueur s'étend du centre du bord gauche du champ de caractère jusqu'au centre du bord droit du champ de caractère.
- 7) Le trait vertical pleine longueur s'étend du centre du bord inférieur du champ de caractère jusqu'au centre du bord supérieur du champ de caractère.
- 8) La représentation graphique typique d'une barre horizontale est un tiret horizontal à peu près au centre d'une lettre majuscule.



Les rectangles entourant les caractères décrits aux notes (1) à (7) ci-dessus sont tracés aux fins d'illustration seulement; ils ne font pas partie intégrante des symboles graphiques.

Tableau 26  
Signes diacritiques

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
·	accent aigu	S 4/2 2/0
`	accent grave	S 4/1 2/0
^	accent circonflexe	S 4/3 2/0
~	tilde	S 4/4 2/0
"	tréma ou Umlaut	S 4/8 2/0
¸	cédille	S 4/11 2/0
•	ogonek	S 4/14 2/0
˘	signe de voyelle brève	S 4/6 2/0
ˇ	accent circonflexe inversé	S 4/15 2/0
¨	accent aigu double	S 4/13 2/0
.	point	S 4/7 2/0
—	signe de voyelle longue	S 4/5 2/0
◌̣	rond	S 4/10 2/0

Note: L'accent grave, l'accent circonflexe et le tilde sont également codés respectivement 6/0, 5/14 et 7/14.

Tableau 27  
Symboles divers sans espacement

GRAPHIQUE	NOM ou DESCRIPTION	REPRÉSENTATION CODÉE
	souligné sans espacement	S 4/12
	barre vectorielle supérieure sans espacement	S 4/10
/	barre oblique sans espacement	S 4/9

**Note:** La représentation graphique de la barre vectorielle supérieure sans espacement est celle d'une flèche vectorielle juste au-dessus de la position d'une lettre majuscule. Lorsque le souligné sans espacement, la barre vectorielle supérieure sans espacement et la barre oblique sans espacement sont utilisés, leur représentation codée précède celle des caractères des tableaux 18 à 26.

Le rectangle entourant le souligné sans espacement est tracé aux fins d'illustration seulement; il ne fait pas partie intégrante du symbole graphique.



## 8. Exigences de conformité

**8.1 Généralités.** La présente syntaxe de données propose un système de codage qui permet de transmettre et d'interpréter correctement des informations spécifiques sans égard aux contraintes matérielles des différentes configurations d'appareil récepteur. Une telle orientation donne aux fabricants et aux fournisseurs d'information toute la souplesse voulue pour mettre au point une vaste gamme de biens et services satisfaisant les exigences des différents secteurs du marché. Toutefois, l'existence même d'une vaste gamme de services est source de problèmes pour le fabricant et pour le fournisseur d'information. Les fabricants d'appareils récepteurs peuvent bien vouloir interpréter et afficher seulement l'information qui est adaptée à leur matériel, tandis que les fournisseurs d'information peuvent bien vouloir que tous les appareils récepteurs soient capables d'afficher uniformément toute l'information. Les exigences de conformité visent à réconcilier ces objectifs dans la mesure du possible.

Les exigences de conformité spécifiées dans le présent document se proposent, de façon générale, de définir les règles (syntaxe) permettant d'obtenir un échange compatible au niveau de l'interface de codage, ainsi que l'exécution (sémantique) normale d'un processus de présentation conforme. Afin d'assurer un échange compatible et fiable dans les deux sens au niveau de l'interface de codage, il est essentiel que tous les processus d'échange se conforment intégralement aux règles de syntaxe définies dans la présente syntaxe de données (voir figure 66). C'est seulement dans le secteur de l'exécution sémantique qu'une vaste gamme de propriétés physiques et de possibilités d'affichage existent, d'où la nécessité de définir des exigences de conformité dans ce secteur.

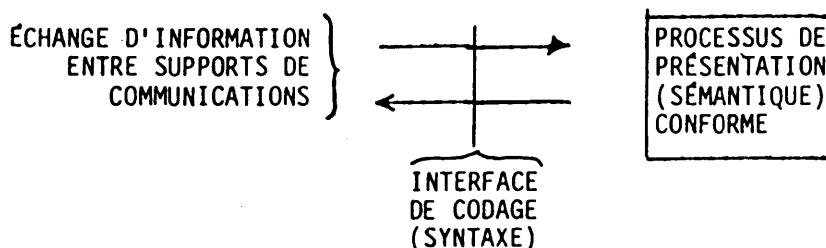


Figure 66

Relation entre entités conformes

Il est reconnu que l'évolution de la technologie et la tendance du marché peuvent déterminer une matérialisation propre à différents secteurs de l'industrie vidéotex/télétexte. Les différentes matérialisations sont appelées modèles de référence de service (MRS). Un MRS général applicable au vidéotex et un MRS général applicable au télétexte sont donnés à l'Annexe D, pour servir de modèles dans la définition de nouveaux MRS.

## 8.2 Conformité de l'échange

8.2.1 Le flot de données formant l'information de la couche présentation échangée par l'interface de codage est assujetti à certaines restrictions syntaxiques afin d'assurer la conformité de l'échange à la présente syntaxe de données. Pour être conforme, l'échange:

- 1) doit comporter une séquence de combinaisons à 7 éléments ou 8 éléments, selon les formats syntaxiques décrits dans le corps de la présente syntaxe de données.
- 2) ne doit pas comporter un multiplet ni une valeur dont la normalisation est remise à plus tard. Cette exigence a pour but de faciliter la compatibilité progressive de toute version éventuelle de la présente syntaxe de données.
- 3) ne doit pas faire jouer un rôle autre que le rôle défini dans la présente syntaxe de données à toute combinaison de bits attribuée par la présente syntaxe de données, sauf si une telle combinaison représente des fonctions ou des caractères d'autres jeux de code appelés par une extension de code d'après la norme ISO 2022-1982, avec l'accord mutuel des parties intéressées.
- 4) ne doit pas comprendre des séquences qui entraîneraient un processus de présentation non défini ni des séquences qui sont définies comme étant erronées dans la présente syntaxe de données; par exemple, une position de dessin ne doit pas être déplacée à l'extérieur de l'écran unitaire.
- 5) doit faire suivre immédiatement les séquences qui produisent un état du processus de présentation lié à la matérialisation de séquences qui ramènent le processus de présentation dans un état connu. Par exemple, le positionnement par suite d'une séquence automatique APR APD, d'une commande de renouvellement de mot et d'une commande d'espacement proportionnel est lié à la matérialisation. Un échange conforme doit rétablir l'emplacement du curseur et de la position de dessin graphique si cette dernière est liée au curseur.

## 8.3 Conformité du processus de présentation

8.3.1 Les définitions ci-dessous s'appliquent à la suite du présent document:

- 1) **Reconnaître** signifie déterminer la forme syntaxique d'une séquence de codage.
- 2) **Exécuter** signifie traiter une séquence de codage pour permettre l'affichage de l'information transmise par la séquence de codage, si elle est présentée, et par les séquences de codage subséquentes, conformément à la présente syntaxe de données (par exemple le calcul des données de position).
- 3) **Présenter** signifie afficher l'information transmise par une séquence de codage et, dans le cas d'une fonction de commande, afficher l'information subséquente touchée par la fonction de commande.

8.3.2 Un processus de présentation de réception conforme à la présente syntaxe de données doit pouvoir recevoir toutes les informations d'échange et reconnaître si ces informations sont conformes ou non (d'après la définition de l'alinéa 8.2). Les informations d'échange non conformes doivent rester sans effet, sauf indication contraire dans la présente syntaxe de données. Cette mesure a pour but de faciliter la compatibilité régressive de toute version éventuelle de la présente syntaxe de données.

8.3.3 Un appareil récepteur d'affichage conforme à la présente syntaxe de données:

- 1) doit exécuter toutes les séquences d'extension de code définies dans la présente syntaxe de données.
- 2) doit exécuter et présenter les 94 caractères du jeu primaire compte tenu des attributs de texte matérialisés.
- 3) doit exécuter tous les autres caractères du répertoire des caractères graphiques (voir 7.1) et les présenter avec exactitude ou sous une forme rapprochée, ou sous la forme d'un symbole (ou de symboles) ne présentant aucun risque de confusion avec tout caractère du répertoire des caractères graphiques.
- 4) doit exécuter et présenter toutes les primitives graphiques géométriques (c'est-à-dire les commandes POINT, LIGNE, ARC, RECTANGLE, POLYGONE et INCRÉMENTATION), doit exécuter et présenter la fonction de commande INITIALISATION, et doit exécuter les fonctions de commande DOMAINE et ATTENTE.
- 5) doit exécuter toutes les valeurs discrètes des fonctions de commande d'attribut qui influent sur la reconnaissance et l'exécution des séquences de codage subséquentes, c'est-à-dire la longueur d'opérande à valeur simple (1 à 4), la longueur d'opérande à valeurs multiples (1 à 8), le mode dimensionnel (2, 3), et les modes de couleur (0, 1 et 2).
- 6) doit exécuter tous les autres codes de commande du jeu IDI et doit les présenter compte tenu des attributs matérialisés.
- 7) doit exécuter et présenter les 65 codes du jeu mosaïque.
- 8) doit exécuter et présenter les 96 macro-instructions, qui sont définis selon les procédures de la présente syntaxe de données.
- 9) doit exécuter et présenter les 96 caractères JCDR, qui sont dynamiquement redéfinissables selon la procédure de la présente syntaxe de données.
- 10) doit exécuter et présenter toutes les fonctions de commande CO définies dans la présente syntaxe de données.
- 11) doit exécuter tous les codes de commande du jeu de code C1 définis dans la présente syntaxe de données, sauf dans le cas des services unilatéraux pour lesquels il se peut que les commandes suivantes ne soient pas nécessaires: CLIGNOTEMENT CURSEUR, CURSEUR STABLE, SUPPRESSION CURSEUR, AVEC PROTECTION et SANS PROTECTION. La présentation des codes de commande C1 est liée à la matérialisation, sauf pour ce qui est des commandes RÉPÉTITION, RÉPÉTITION JUSQU'À EOL, AVEC DÉFILEMENT et SANS DÉFILEMENT, qui doivent être exécutées et présentées.

**8.3.4** Dans un appareil récepteur d'affichage conforme à la présente syntaxe de données, les attributs peuvent être matérialisés des différentes façons ci-dessous.

(1) Tous les paramètres des commandes ou des attributs de dessin qui sont spécifiés sous forme de variables continues peuvent être exprimés de façon rapprochée par des valeurs liées à la matérialisation, l'une des valeurs étant la valeur par défaut.

(2) Tous les paramètres des commandes ou attributs de dessin qui peuvent prendre des valeurs discrètes multiples (deux ou plus) peuvent être exprimés de façon rapprochée par des valeurs liées à la matérialisation, l'une des valeurs étant la valeur par défaut.

(3) Pour l'interprétation de tous les échanges conformes à la présente syntaxe de données, la position du curseur et la position de dessin doivent être calculées de façon à minimiser l'accumulation des erreurs d'arrondi.

**8.3.5** Les conditions par défaut des attributs de tous les processus de transmission et de réception de la couche présentation conformes à la présente syntaxe de données sont résumées au tableau 28.

**Tableau 28**  
**Conditions par défaut**

Paramètres	Valeurs
(a) Conditions par défaut à l'initialisation et conditions rétablies par la commande NSR.	
Tableau effectif et jeux actifs C et G:	Comme sur les figures 4 et 6
Mode de couleur:	0
Couleur de dessin:	Blanc
Longueur pour valeur simple (adresse de la carte des couleurs):	1 multiplet
Longueur pour valeurs multiples (coordonnées):	3 multiplets
Mode dimensionnel:	2 dimensions avec $Z = 0$
Taille du pel logique:	$dx = 0, dy = 0$ (origine au coin inférieur gauche)
Rotation de texte:	$0^\circ$ (par rapport à l'horizontale)
Trajet de caractère:	Vers la droite
Espacement des caractères:	1 (largeur du champ de caractère actuel)
Espacement des rangées:	1 (hauteur du champ de caractère actuel)
Paramètres de déplacement:	Le curseur et la position de dessin graphique se déplacent ensemble
Emplacements du curseur et de la position de dessin:	Non définis
Style du curseur:	Soulignement
Affichage curseur:	Suppression (invisible)
Dimensions du champ de caractère:	$dx = 1/40, dy = 5/128$ , compte tenu de la résolution physique
Texture de ligne:	Pleine
Rehaussement:	Sans rehaussement
Motif:	Plein
Dimensions du masque de texture:	$dx = 1/40, dy = 5/128$ , compte tenu de la résolution physique
Champ actif:	Écran unitaire
Mode soulignement:	Sans soulignement
Mode renouement de mot:	Sans renouement
Mode défilement:	Sans défilement
Mode vidéo inversée:	Sans vidéo inversée
(b) Autres conditions par défaut à l'initialisation.	
Carte des couleurs:	Échelle des demi-tons de gris, nuances demi-saturées, d'après l'alinéa 5.3.2.5.2
Condition de clignotement:	Sans clignotement
Macro-instructions:	Néant
JCOR:	Néant
Motifs programmables:	Néant
Champs sans protection:	Non définis

### 9. Possibilités de développement

Des possibilités de développement sont prévues par les techniques d'extension de code définies dans la présente syntaxe de données. Les nouvelles possibilités pourraient être regroupées en classes logiquement homogènes permettant de définir de nouveaux jeux de code complets à ajouter au répertoire existant. L'adjonction de nouvelles fonctions aux jeux G et C décrits dans la présente syntaxe de données, en plus de ce qui est déjà défini (par exemple le cas des commandes IDI qui sont spécifiées par des séquences fixes ou des commandes erronées dont l'appareil récepteur ne doit pas tenir compte), fera éventuellement l'objet d'une extension compatible avec la présente syntaxe de données.

## Annexe A

### Modèle d'architecture en couches

Le modèle de référence de base pour l'interconnexion des systèmes ouverts est décrit dans la norme ISO 7498-1983. Le principe d'un modèle semblable est également défini dans l'Avis (provisoire) X.200 du CCITT. L'architecture en couches est un système de protocoles connexes essentiels à la définition d'un système de communication complet. L'architecture en couches permet "l'interconnexion des systèmes ouverts" (c'est-à-dire l'échange des données) entre systèmes participants. Chaque couche tient compte d'un aspect indépendant d'un système de communication, ce qui permet les substitutions de protocoles, aux différentes couches, selon la nature des moyens utilisés. La figure A1 illustre le modèle. La structure de codage décrite dans la présente syntaxe de données porte sur les données fournies par l'utilisateur au niveau de la couche présentation du modèle de référence de base; elle est indépendante des protocoles utilisés pour transférer les données entre systèmes.

Les services vidéotex et télétexte utilisent la même syntaxe et la même sémantique de protocole de la couche présentation mais il se peut qu'ils utilisent des protocoles différents aux autres couches. Les protocoles des autres couches applicables au télétexte sont définis dans le Rapport 957 du CCIR, "Caractéristiques des systèmes télétexte", document 11/5001-E, octobre 1981; dans le Cahier des charges sur la radiodiffusion CR-14, publié par le Service de réglementation des télécommunications du ministère des Communications, juin 1981; et dans la North American Basic Teletext Specification (NABTS).

Les sept couches peuvent être divisées en deux groupes. Les couches 1 à 4 se rapportent à la transmission des données et les couches 5 à 7, au traitement et à l'utilisation de ces données.

(1) La couche physique assure des fonctions ayant trait aux aspects mécaniques, aux aspects électriques et aux procédures afin d'établir, de maintenir et de libérer des connexions physiques.

(2) La couche liaison fournit une liaison de transmission des données par une ou plusieurs connexions physiques. La correction des erreurs, le séquençement et la commande de débit sont effectués afin de maintenir l'intégrité des données.

(3) La couche réseau assure des fonctions d'acheminement, de commutation et d'accès au réseau de façon que la couche transport ne soit pas informée de la manière dont sont utilisées les ressources de transmission sous-jacentes.

(4) La couche transport fournit un circuit virtuel transparent de transmission des données de bout en bout sur une ou plusieurs installations de réseau en tandem.

(5) La couche session permet d'établir une connexion de session et aide à l'échange ordonné des données ainsi qu'à l'exécution d'autres fonctions de commande connexes s'appliquant à un service de communication particulier.

(6) La couche présentation permet de représenter l'information codée de façon que sa signification ne se perde pas. Les formats de codage détaillés du système décrit dans le présent document constituent la base de la syntaxe des données de protocole de la couche présentation du vidéotex, du télétexte et des applications connexes.

(7) La couche application est la couche supérieure du modèle de référence. Les protocoles de cette couche fournissent les services désirés par l'utilisateur final. À titre d'exemple, les commandes du service de recherche d'information d'une application vidéotex font partie de la couche application.



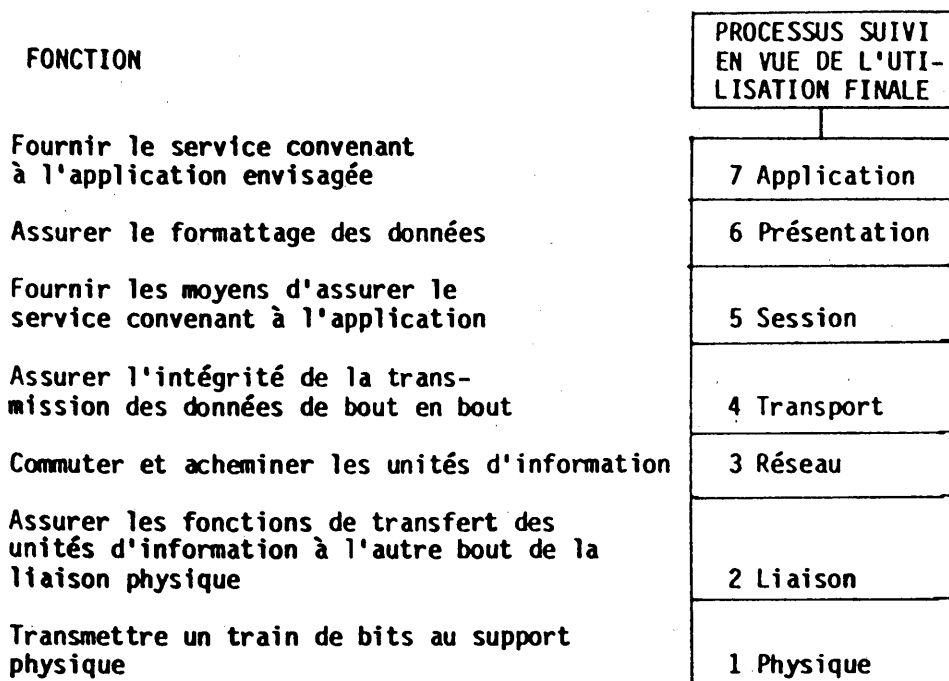


Figure A1

Les sept couches, classées selon leurs fonctions,  
du modèle d'interconnexion des systèmes ouverts (OSI)

## Annexe B

### Principe du système de coordonnées

La syntaxe de données fait appel au principe du système abstrait de coordonnées cartésiennes permettant de définir toutes les opérations de dessin d'image, de texte et d'incrémentation. Les opérations font appel à des coordonnées unitaires normalisées. En mode bidimensionnel, le plan de dessin est formé d'une plage carrée, l'écran unitaire, dont les points sur l'axe des X et des Y s'étendent de 0 (inclusivement) à 1 (exclusivement). En mode tridimensionnel, l'espace de dessin est formé par un cube dont les points sur l'axe des X, Y et Z s'étendent de 0 à 1, le point 0 étant le point le plus éloigné de l'observateur sur l'axe des Z. Aucun autre détail de ce mode tridimensionnel n'a encore été défini.

Les commandes de dessin et les autres opérations font appel à des spécifications de coordonnées, qui sont des fractions binaires de l'écran unitaire. Les spécifications de dessin à l'extérieur de l'écran unitaire, directement déterminées dans l'espace des coordonnées négatives au-dessous ou à la gauche du zéro de l'écran unitaire, ou déterminées par déplacement relatif à l'extérieur de l'écran unitaire, sont erronées.

Les spécifications de coordonnées peuvent être décrites à plusieurs niveaux de précision en raison de leur représentation sous forme de fractions dans la présente syntaxe de données. Ces fractions sont écrites dans les termes abstraits de la zone de dessin unitaire seulement, afin de n'éliminer aucune matérialisation possible. Toutefois, ces fractions "naturelles" peuvent ne pas être exactement représentables par l'arithmétique "interne" de l'appareil récepteur, d'où la nécessité d'une certaine approximation. (Exemple:  $1/40$  est une fraction irrationnelle en représentation binaire). Le résultat voulu s'obtient par élimination des bits de poids faible non représentables, grâce à un processus de troncation lié à la matérialisation. Lorsque le MRS spécifie une résolution physique de 256 pixels, une façon commode d'obtenir une représentation du texte et des graphiques, "compte tenu de la résolution physique", consiste à maintenir au moins 12 bits de précision dans l'appareil récepteur. Par exemple, la largeur de caractère  $1/40$  du TEXTE NORMAL est représentée par  $102/4096$  à 12 bits de précision. À noter que pour des résolutions physiques supérieures, une précision interne supérieure correspondante peut devenir nécessaire pour maintenir la compatibilité avec la résolution physique.

Il existe en principe plusieurs représentations numériques utilisables dans le processus de communication. Il y a d'abord la représentation de précision infinie théorique ou "naturelle" dans laquelle les images sont définies. Cette représentation doit se trouver dans les limites de la syntaxe de la présente syntaxe de données, c'est-à-dire avec un DOMAINE spécifié pour les communications.

Dans un appareil récepteur, cette représentation numérique correspond à la précision "interne" de l'appareil. Il est possible qu'un certain degré de précision soit perdu à cette étape-ci. Les coordonnées (ou positions) sont maintenues avec cette résolution interne. Le dessin réel fait appel à la résolution physique de l'affichage. Il y a transposition de la représentation interne à la représentation physique, ce qui peut entraîner une autre perte de précision.

Les pixels physiques excités sont ceux auxquels correspond toute partie de la représentation interne du dessin; par exemple, une représentation interne d'une précision de 12 bits pourrait correspondre à une résolution physique de 256

pixels (8 bits). La représentation dans ce cas comporterait simplement une troncation de 4 bits de poids faible. Lorsque les représentations ne sont pas simplement reliées par une puissance de 2, la transposition devient plus complexe.

La présente syntaxe de données ne spécifie pas intentionnellement les relations entre les trois représentations; elle stipule simplement qu'elles doivent être compatibles. Cette mesure vise à permettre que différentes matérialisations respectent la compatibilité spécifiée dans le MRS.

La syntaxe de données ne tient compte que des termes abstraits de l'écran unitaire afin de n'éliminer aucune matérialisation possible; toutefois, une telle abstraction peut entraîner une certaine confusion pour le concepteur et pour le fournisseur d'information. Étant donné que le modèle de référence d'un service particulier définit certains paramètres comme la résolution, il devient possible de bien clarifier les résultats de l'affichage qui serait obtenu sur différents types de terminaux. Un exemple est donné ci-dessous pour montrer les paramètres de l'affichage physique résultant du choix d'une matérialisation comportant une mémoire à plans de bits, avec 256 pixels sur l'axe des X et des plans de 4 bits.

La syntaxe de données indique que le cadrage obtenu par les techniques de dessin d'image, de texte et de photographie est garanti seulement dans certaines limites. Cela s'explique du fait que pour certaines résolutions ou avec certaines technologies d'affichage, le nombre de pixels physiques d'affichage ne se divise pas exactement pour correspondre au nombre de pixels logiques spécifiés, c'est-à-dire qu'il peut en résulter une erreur de quantification. L'erreur de quantification produit un effet semblable aux configurations moirées en optique, ce qui peut entraîner une grave altération de l'image.

Des bords dentelés semblables à ceux qui se produisent dans le dessin des segments de lignes peuvent se produire dans le dessin des caractères ou ailleurs, principalement en raison des conversions d'échelle. Différentes techniques de matérialisation permettent d'éliminer ou de minimiser cet effet de dentelé.

Examinons par exemple le cas de 256 pixels physiques de résolution sur chacun des axes d'une mémoire à plans de bits: l'écran unitaire de 0 (inclusivement) à 1 (exclusivement) en donne une correspondance exacte. La représentation binaire des fractions correspond exactement à la position des pixels, ce qui donne une ligne de 45 degrés tout à fait lisse. Les pixels doivent être carrés ou très proches d'une forme carrée pour éviter toute apparence de déformation géométrique. Les cercles doivent paraître ronds. Cela signifie que dans le sens vertical les pixels physiques devraient s'aligner avec les lignes de balayage de l'affichage télévisuel.

Dans les MRS vidéotex et télétexte, les dimensions de texte par défaut garantissent 40 caractères par rangée et 20 rangées dans la plage d'affichage d'un écran de télévision. Le rapport de forme s'applique aux limites extérieures de l'écran d'affichage physique. Lorsque l'extrémité de réception comporte un décodeur distinct du récepteur de télévision, il convient de prévoir le surbalayage dans la conception du décodeur. Voir le document "Society of Motion Picture and Television Engineers (SMPTE) Recommended Practice, Specifications for Safe Action and Safe Title Areas, Test Pattern for Television Systems, SMPTE RP 27.3-1972".

Le nombre 256 ne se divise pas exactement par 40 et, lorsqu'une largeur de champ de caractère de  $1/40$  de l'écran unitaire est définie, il peut se produire des effets dus à l'erreur de quantification. Par exemple, les 40 positions de caractère pourraient être définies de façon à correspondre à un peu moins de 256 pixels de l'écran d'affichage pour minimiser les effets dus à l'erreur de quantification.

Des études psychologiques ont montré que les caractères les plus faciles à lire dans cette plage de représentation comportent 6 pixels par 10 pixels. Cela signifie donc que dans le sens vertical 200 lignes de balayage sont nécessaires pour produire 20 rangées.

Si les images sont représentées par pixels à l'aide de la commande INCRÉMENTATION POINT, elles ne peuvent pas faire l'objet d'une conversion d'échelle sans entraîner une erreur de quantification, sauf par division ou multiplication exacte d'un nombre entier simple. Cela signifie que de telles images peuvent être réduites dans un processus de transformation d'un appareil comportant une certaine densité d'affichage physique à un autre appareil comportant une résolution différente sans relation simple. Pour une raison semblable, la conversion d'échelle des caractères JCDR et des masques de texture définissables se limite à une division ou une multiplication exacte d'un nombre entier. Une telle conversion d'échelle est nécessaire lorsqu'un caractère JCDR ou un masque de texture est décrit par un champ de caractère ou des dimensions dont la valeur est supérieure à la limite physique imposée par la capacité de mémoire.

Les paramètres physiques de l'appareil d'affichage utilisé dans notre exemple sont donnés ci-dessous.

- (1) 256 pixels sur l'axe des X par 200 pixels sur l'axe des Y.
- (2) Rapport de forme de 0.78125.
- (3) Dimensions physiques du champ de caractère par défaut de  $102/4096$  sur l'axe des X par  $10/256$  sur l'axe des Y.

Des dimensions de caractère autres que celles du texte NORMAL par défaut peuvent également s'adapter à cette plage d'affichage. Un texte de 32 positions de caractère et de 16 rangées se divise exactement en 256 positions de pixels sur l'axe des X et fait appel à 192 lignes de balayage sur l'axe des Y, ce qui entraîne des dimensions physiques du champ de caractère de  $12/256$  sur l'axe des Y par  $8/256$  sur l'axe des X. Un texte de 40 positions de caractère et de 24 rangées donnerait  $102/4096$  sur l'axe des X par  $8/256$  sur l'axe des Y.

Examinons un autre exemple: 240 pixels physiques de résolution peuvent être utilisés dans toute la plage de 0 à 1 sur l'axe des X, avec 200 pixels sur l'axe des Y. Le rapport de forme de 0.78125 reste le même que dans l'exemple précédent. Les dimensions physiques du champ de caractère par défaut sont de  $6/240$  sur l'axe des X par  $10/256$  sur l'axe des Y. Cela donne exactement 40 colonnes de caractères. Il se peut que, dans le cas de 240 pixels physiques, la transformation des spécifications de fractions binaires des coordonnées ne soit pas exacte et entraîne une erreur de quantification.

Étant donné que les nombres utilisés dans les exemples ci-dessus peuvent s'appliquer aux MRS, ils peuvent être utilisés pour aider les fournisseurs d'information à définir les images.

## Annexe C

### Extension de code et transformation 7 éléments/8 éléments

La norme ISO 2022-1982, "Jeux ISO de caractères codés à 7 et à 8 éléments" - Techniques d'extension de code", spécifie la structure du codage des caractères à 7 et à 8 éléments (voir 4.3). L'extension de code désigne les façons de représenter un plus grand nombre de caractères graphiques et de fonctions de commande que le nombre respectivement obtenu de 128 ou 256 en codage 7 ou 8 éléments. Il existe deux moyens d'extension: 1) changer l'interprétation du jeu C ou G en appelant un nouveau jeu, et (2) représenter un caractère graphique ou une fonction de commande sous forme d'une séquence de multipliets. Le premier caractère d'une telle séquence s'appelle un introducteur car il change l'interprétation des multipliets qui suivent. Chaque introducteur comporte une syntaxe spécifiée pour les multipliets qui suivent.

Dans la syntaxe de données, les séquences du tableau C1 servent à représenter des caractères graphiques ou des fonctions.

Tableau C1

Nom de l'introducteur	Exemples de syntaxe	Renvoi
ÉCHAPPEMENT	ESCI ... IF	4.3.2
SS2, SS3	SS2 G	4.3.2
Accent sans espacement	G G	5.2
Code d'opération IDI	P D ... D	5.3
APS	APS G G	6.1.2.4
NSR	NSR G G	6.1.6.5
DÉF MACRO, DÉFP MACRO, DÉFT MACRO, DÉF JCDR	DÉF MACRO M ... FIN	6.2.2
DÉF TEXTURE		
RÉPÉTITION	RÉPÉTITION G	6.2.7.2

La syntaxe de données spécifie également un processus de correction d'erreur lorsque se présente un caractère inattendu dans une séquence. La correction consiste à traiter la séquence partielle comme une opération vide et à exécuter le caractère inattendu.

La norme ISO 2022-1982 spécifie un algorithme permettant les transformations 7 éléments/8 éléments, compte tenu de certaines conditions minimales des jeux de caractères codés à transformer. Ces transformations sont typiquement exécutées aux centres têtes de ligne entre services. Les conditions sont données ci-dessous.

- (1) Les jeux C se trouvent aux colonnes 0-1 et 8-9.
- (2) Les jeux G se trouvent aux colonnes 2-7 et 10-15.
- (3) Dans le jeu C0, les commandes ESC, S0 et SI sont respectivement codées par 1/11, 0/14 et 0/15.

(4) Les inversions avec verrouillage supplémentaires sont codées par ESC 6/14, ESC 6/15, ESC 7/14, ESC 7/13 et ESC 7/12.

(5) Tout autre caractère C0 et C1 peut devenir un introducteur.

La plus simple transformation de 7 éléments à 8 éléments ajoute simplement b8 = 0 aux données. Cela signifie que GR est inutilisé et que les commandes C1 sont représentées sous forme de séquences à 2 multipliets ESC Fe.

Considérons l'exemple ci-dessous de transformation selon la norme ISO 2022.

7 éléments: SI 4/1 S0 2/1 SS2 3/1

La transformation vers 8 éléments place G0 dans GL et G1 dans GR:

8 éléments: 4/1 10/1 SS2 11/1

Dans le multipliet qui suit SS2, le bit b8 est remis à 1 étant donné que le flot de données est toujours dans l'état hors code (G1). (L'algorithme de transformation ne tient pas compte de SS2 ni de tout autre introducteur, sauf ESC). Pour cette raison, la norme spécifie que le 8<sup>e</sup> bit, b8, doit rester sans effet dans les multipliets d'opérande qui suivent un introducteur.

**Note:** Dans le cas des multipliets qui suivent le multipliet d'opérande simple après les DÉF jusqu'à FIN, le bit b8 est significatif.

Considérons l'exemple ci-dessous dans lequel G0 est placé dans GL et G1 est placé dans GR:

8 éléments: 4/1 10/1 SS2 3/1

Après transformation vers 7 éléments:

7 éléments: 4/1 S0 2/1 SS2 SI 3/1

L'algorithme de transformation place un caractère SI avant 3/1 étant donné que le flot de données de 8 bits est passé de G1 (GR) à G0 (GL). De cette façon, une inversion avec verrouillage a été introduite entre l'introducteur et son multipliet d'opérande. Les auteurs de la norme considèrent qu'il est trop complexe d'exiger qu'un appareil récepteur soit capable de traiter les inversions avec verrouillage entre un introducteur et ses données d'opérande à un ou deux multipliets, surtout parce que de nombreuses inversions avec verrouillage sont elles-mêmes constituées par des séquences d'échappement à deux multipliets.

Un algorithme préférable de transformation de 8 éléments à 7 éléments consisterait à traiter tous les caractères C0 et C1 comme s'ils étaient des introducteurs et de toujours insérer une inversion avec verrouillage avant tout caractère du jeu C suivi d'un caractère du jeu G nécessitant une inversion. Cette solution suppose que les algorithmes de transformation de 8 éléments à 7 éléments mettent en tampon un caractère d'anticipation.

## Annexe D

### Modèle de référence de service (MRS) général du vidéotex et modèle de référence de service (MRS) général du télétexte

#### Domaine d'application

Les modèles de référence de service (MRS) définissent des ensembles d'exigences de matérialisation propres aux services vidéotex et télétexte. Ces ensembles d'exigences représentent les fonctions maximales que le fournisseur d'information doit supposer pour le codage de l'information de texte et d'image. Ces ensembles représentent également les fonctions minimales que le fabricant des appareils récepteurs d'affichage devrait matérialiser afin d'assurer l'affichage satisfaisant de l'information de texte et d'image. Le matériel et les échanges d'information satisfaisant aux exigences de ces MRS doivent également satisfaire aux exigences de conformité de la syntaxe de données.

Les MRS sont combinés au tableau D1. Les exigences sont identiques dans pratiquement tous les cas. Lorsqu'il y a divergence, les paramètres particuliers des services vidéotex et télétexte sont spécifiés séparément. Les différences mineures s'expliquent principalement par la nature unidirectionnelle du service télétexte par comparaison à la nature bidirectionnelle du service vidéotexte. Certaines caractéristiques, comme le champ SANS PROTECTION, sont significatives seulement dans le cas du service vidéotexte; elles ne sont donc pas utilisées dans le service télétexte.

Les exigences données au tableau D1 définissent les fonctions garanties (par un appareil récepteur d'affichage qui satisfait aux exigences du MRS applicable au fournisseur d'information et à l'exploitant d'un service pour la génération et l'affichage de l'information. En temps normal, il est entendu que le fournisseur d'information crée l'information et que le fabricant produit des appareils récepteurs d'affichage conformes à ces exigences. Toutefois, les variantes ci-dessous peuvent se présenter.

(1) -Le fournisseur d'information peut créer de l'information qui dépasse les exigences du MRS applicable. Dans ce cas, un appareil récepteur d'affichage qui satisfait aux exigences du MRS pourrait normalement afficher les informations supplémentaires de façon raisonnablement satisfaisante. Toutefois, un tel affichage n'est pas toujours garanti. Par exemple, le fournisseur d'information peut raisonnablement faire appel à une palette de 4096 couleurs si ces couleurs sont sélectionnées de façon que les appareils récepteurs d'affichage ne comportant qu'une palette de 512 couleurs donnent un affichage satisfaisant.

(2) Un appareil récepteur d'affichage peut dépasser les exigences du MRS applicable. Un tel appareil peut produire un affichage plus agréable indépendamment des exigences supplémentaires associées à l'information créée par le fournisseur. Par exemple, un appareil récepteur d'affichage dont la résolution horizontale est de 512 pixels peut produire des images plus agréables qu'un appareil récepteur dont la résolution horizontale est de 256 pixels, même si le fournisseur d'information fait appel à une plage d'adresses de seulement 256 pixels.

(3) Un appareil récepteur d'affichage peut matérialiser des caractéristiques à un degré inférieur à celui qui est défini dans le modèle de référence de service et se conformer quand même à la syntaxe de données, de la même façon qu'un téléviseur monochrome se conforme à la norme de télévision en couleurs établie par le National Television System Committee (NTSC). Il y a lieu de noter que dans un tel cas une partie de l'information peut se perdre, tout comme un téléviseur monochrome perd l'information de couleur. De façon analogue, dans les deux cas ces normes ont été organisées de façon que l'essentiel de l'information soit préservé.

Les paramètres donnés au tableau D1 ont été identifiés comme étant nécessaires pour définir un MRS du vidéotex et un MRS du télétexte unidirectionnel. La section de la syntaxe de données portant sur les exigences de conformité indique les fonctions qui doivent être exécutées et présentées avec exactitude ainsi que les fonctions qui peuvent être présentées sous forme rapprochée.

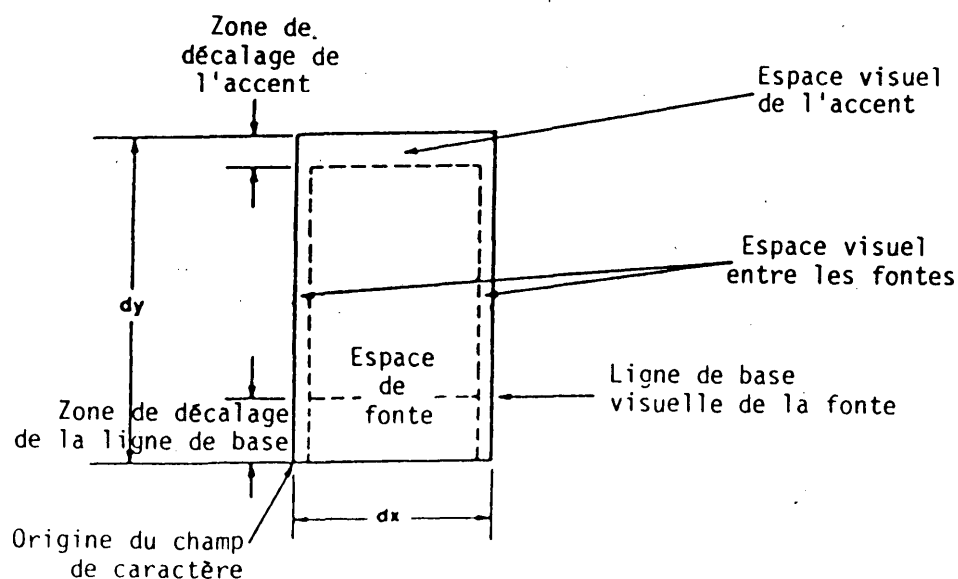


Figure D1  
Plan du champ de caractère



Tableau D1

Modèles de référence de service (MRS) généraux du vidéotex et du télétexte

Fonctions	Exigences
1. Séquences d'extension de code	Toutes les séquences d'extension de code définies dans la syntaxe de données (compte tenu des couches inférieures du service) doivent être exécutées (voir 8.3.3 (1) ).
2. Jeu de caractères primaire	<p>Les 94 caractères primaires doivent être présentés de façon unique dans un champ de caractère aux dimensions supérieures ou égales à <math>dx = 6/256</math>, <math>dy = 8/256</math> (voir 8.3.3 (2)).</p> <p>La zone de décalage de la ligne de base visuelle de la fonte, c'est-à-dire la ligne inférieure apparente des caractères alphabétiques majuscules (voir figure D1) doit correspondre à 20 % de la hauteur <math>dy</math> de la dimension du champ de caractère actuel, compte tenu de la résolution physique.</p> <p>L'espace de l'accent (voir figure D1) doit être suffisant pour produire un affichage discernable, mais pas nécessairement distinguable.</p> <p>Il est recommandé, dans la mesure du possible, que la fonte soit centrée à l'intérieur du champ de caractère, laissant entre les fontes un espace visuel égal à la gauche et à la droite de chacune; dans tous les cas, un certain espace doit être prévu à la droite de chaque fonte.</p> <p>Tous les autres aspects des fontes de caractère sont liés à la matérialisation.</p>
3. Tous les autres caractères du répertoire de caractères graphiques	<p>Tous les autres caractères du répertoire de caractères graphiques (voir 7.1) doivent être présentés.</p> <p>Les fontes de caractère doivent être visuellement conformes à celles du jeu de caractères primaire, mais elles sont liées à la matérialisation (voir 8.3.3 (3)).</p>

---

4. Jeu IDI, primitives géométriques de dessin	Toutes les primitives géométriques de dessin doivent être exécutées et présentées de la façon spécifiée en 8.3.3 (4).  Le nombre de sommets d'un polygone et d'un tracé curviligne irrégulier doit se limiter à 256. (voir 5.3.3.3.1, 5.3.3.5.1 et 5.3.3.6.5).
5. Jeu IDI, fonctions de commande d'attribut	Toutes les fonctions de commande d'attribut doivent être exécutées de la façon spécifiée en 8.3.(4), (5) et (6), et présentées comme ci-dessous.
(1) DOMAINE	
(a) Longueur d'opérande à valeur simple	Toutes les longueurs d'opérande à valeur simple doivent être exécutées.
(b) Longueur d'opérande à valeurs multiples	Toutes les longueurs d'opérande à valeurs multiples doivent être exécutées.
(c) Mode dimensionnel	Les modes bidimensionnel et tridimensionnel doivent être exécutés et présentés. Toutefois, la troisième dimension doit rester sans effet, le cas échéant.
(d) Pel logique	La gamme complète des hauteurs et largeurs de pel logique, compte tenu de la résolution physique, doit être exécutée et présentée.
(e) Exécution du dessin	L'emplacement de la position de dessin et du curseur doit être maintenu, compte tenu de la résolution physique (voir article 10 du présent tableau et Annexe B).
(2) TEXTE	
(a) Rotation de caractère	Les quatre orientations (c'est-à-dire 0°, 90°, 180° et 270°) doivent être exécutées et présentées.
(b) Trajet de caractère	Les quatre sens (c'est-à-dire vers la gauche, vers la droite, vers le haut et vers le bas) doivent être exécutés et présentés.

---

---

(c) Espacement des caractères	Les quatre espacements (c'est-à-dire 1, 5/4, 3/2 et espacement proportionnel) doivent être exécutés et présentés, compte tenu de la résolution physique. L'espacement proportionnel est lié à la matérialisation, étant donné que la fonte n'est pas définie.
(d) Espacement des rangées	Les quatre espacements (c'est-à-dire 1, 5/4, 3/2 et 2) doivent être exécutés et présentés, compte tenu de la résolution physique.
(e) Attributs de déplacement	Les quatre attributs de déplacement (c'est-à-dire ensemble, curseur en avance, position de dessin en avance et indépendamment) doivent être exécutés.
(f) Style du curseur	Les quatre styles (c'est-à-dire soulignement, bloc, réticule et spécial) doivent être exécutés et présentés.
(g) Dimensions du champ de	Les dimensions du champ de caractère supérieures ou égales à $dx = 6/256$ , $dy = 8/256$ (voir Annexe B) doivent être exécutées et présentées compte tenu de la résolution physique, y compris les formats d'affichage de caractère ci-dessous (colonne x rangée): 40 x 24, 40 x 20, 40 x 10, 32 x 16, 20 x 10.
(3) TEXTURE	
(a) Texture de ligne	Les quatre textures (c'est-à-dire pleine, pointillée, tiretée et mixte) doivent être présentées.
(b) Motif	Les quatre masques définis (c'est-à-dire plein, hachures verticales, hachures horizontales et hachures croisées) doivent être présentés. Les quatre masques programmables doivent être présentés à un niveau de résolution de 16 x 16 éléments mémorisés.
(c) Rehaussement	L'attribut de rehaussement doit être exécuté de la façon spécifiée en 5.3.2.4.3 dans tous les modes de couleur.

---

---

(4) COULEUR	<p>Tous les modes de couleur (c'est-à-dire 0, 1 et 2) doivent être présentés. Seize couleurs simultanées, parmi un jeu de 512 couleurs obtenues par attribution de 3 bits respectivement à V, R et B, doivent être disponibles. Les valeurs exactes des couleurs par défaut ne sont pas garanties, en raison des erreurs d'arrondi de l'équation des couleurs (voir 5.3.2.5.2).</p> <p>La référence de chrominance et de luminance des couleurs primaires est celle qui est définie par le système NTSC (FCC Rules and Regulations, Part 73 Radio Broadcast Services, Subpart E).</p>
(5) CLIGNOTEMENT	<p>Les commandes CLIGNOTEMENT du jeu IDI et du jeu C1 doivent être exécutées et présentées. Seize processus de clignotement doivent être disponibles.</p>
(6) ATTENTE	<p><u>Vidéotex</u>: La durée d'un intervalle d'attente minimale (opérande zéro, voir 5.3.2.8.2) doit se situer entre 0s et 1/10 s inclusivement.</p> <p><u>Télétexte</u>: La durée d'un intervalle d'attente minimal (opérande zéro, voir 5.3.2.8.2) doit se situer entre deux champs de télévision complets (approximativement 1/30 s) et trois champs de télévision complets inclusivement.</p>
(7) INITIALISATION	<p>Cette commande doit être exécutée et présentée de la façon décrite en 5.3.2.9.</p>
6. Jeu mosaïque	<p>Tous les caractères mosaïques distincts et contigus doivent être présentés de façon unique dans un champ de caractère aux dimensions égales ou supérieures à <math>dx = 6/256</math>, <math>dy = 8/256</math> (voir 8.3.3 (7)).</p>
7. Jeu de macro-instructions	<p>Il doit être possible de définir 96 macro-instructions, compte tenu des limitations de mémoire. Les macro-instructions peuvent être mises en jeu par un mécanisme d'entrée de</p>

---

---

l'utilisateur, par exemple une touche. Les macro-instructions de transmission qui sont associées à un mécanisme d'entrée de l'utilisateur peuvent toujours être mises en jeu. Les autres macro-instructions qui sont associées à un mécanisme d'entrée de l'utilisateur ne sont disponibles que si le curseur est présent. Le macro-code mis en jeu par l'utilisateur peut entraîner un état du processus de présentation de la réception lié à la matérialisation. Un macro-code mis en jeu de cette façon doit être exécuté selon une séquence contiguë jusqu'à la fin. Le fournisseur de service doit ramener l'appareil récepteur à un état connu, le cas échéant.

Vidéotex: Un minimum de huit associations de ce genre est nécessaire pour mettre en jeu les macro-instructions 2/0 à 2/7. Il est recommandé que le fournisseur de service définisse les macro-instructions sans association à partir de 7/15. Une association peut être représentée, par exemple, par une touche particulière ou une séquence de manipulation. Une macro-instruction de transmission doit être considérée comme une communication vers un ordinateur central (voir 8.3.3 (8)).

Télétexte: Une macro-instruction de transmission doit être considérée comme une transmission vers un processeur local (voir 8.3.3 (8)).

## 8. JCDR

Il doit être possible de définir 96 caractères JCDR, compte tenu des limitations de mémoire. La résolution minimale du tampon de mémoire doit être la résolution physique minimale (comme il est spécifié par l'article 10 du présent tableau) associée au champ de caractère au début de la définition JCDR. Par exemple, si la taille du champ de caractère est  $dx = 6/256$  et  $dy = 10/256$ , le tampon de mémoire doit être formé d'un réseau d'au moins 6 éléments horizontaux et

---

---

	10 éléments verticaux. Cette résolution minimale est supposée par les exigences de mémoire de l'article 11 du présent tableau. Les attributs de TEXTE définis par l'article 5 (2) du présent tableau doivent s'appliquer durant la définition et l'affichage JCDR (voir 8.3.3 (9)).
9. Jeu de COMMANDE CO	<p>Ce jeu doit être exécuté et présenté de la façon décrite en 8.3.3 (10).</p> <p><u>Vidéotex</u>: Les couches inférieures du service ne garantissent pas que le caractère CAN sera traité, (voir 8.3.3 (10)) sauf si le caractère CAN suit immédiatement l'appel d'une macro-instruction.</p> <p><u>Télétexte</u>: Les couches inférieures du service ne garantissent pas que le caractère CAN sera traité (voir 8.3.3 (10)).</p>
10. Paramètres de l'affichage physique	<p>La résolution doit être de l'ordre de 256 pixels horizontalement et 200 pixels verticalement. La pleine dimension X (largeur) de l'écran unitaire et la dimension Y (hauteur) de 0.0 à 0.78125, doivent être visibles dans la plage d'affichage. La matérialisation de la plage périphérique n'est pas garantie (voir 4.2.2).</p>

---

---

11. Exigences de mémoire

L'appareil récepteur doit comporter au moins 32 multiplets de mémoire organisés sous forme de 16 x 16 pixels pour chaque masque de texture programmable. Cette capacité de mémoire ne tient pas compte des totaux décrits ci-dessous.

Vidéotex: La mémorisation globale des définitions de macro-codes, des caractères JCDR et des champs sans protection ne doit pas nécessiter plus de 3 kilo-multiplets (3072 multiplets). Ce total ne tient pas compte de la servitude de matérialisation dans l'appareil récepteur, et suppose les conditions ci-dessous.

a) La mémorisation des macro-codes nécessite un multiplet pour chaque multiplet de définition de code, sauf dans le cas des codes DÉF et de terminaison.

b) Trois multiplets de mémorisation sont nécessaires pour chaque cellule de caractère complète dans les champs sans protection.

c) La mémorisation du jeu JCDR doit permettre de définir 96 caractères de taille NORMALE dans la mémoire partagée tout en laissant deux kilo-multiplets (2048 multiplets) de cette mémoire pour les macro-codes et les champs sans protection, pourvu que le total du code de la couche présentation définissant les caractères JCDR ne dépasse pas trois kilo-multiplets (3072 multiplets).

Télétexte: La mémorisation totale des macro-codes et des caractères JCDR ne doit pas nécessiter plus de 3 kilo-multiplets (3072 multiplets). Ce total ne tient pas compte de la servitude de matérialisation dans l'appareil récepteur. Les conditions sont identiques à celles posées pour le MRS du vidéotex, mais ce cas est traité avec d'autres limitations de mémoire dans la norme NABTS.

---

---

12. Jeu de COMMANDE C1

Ce jeu exécute et présente les codes suivants: TEXTE NORMAL, TEXTE MOYEN, DOUBLE HAUTEUR, DOUBLE TAILLE (voir 8.3.3(11))). Les caractères qui suivent un code de commande TEXTE PETIT (4/10) doivent être exécutés sans être nécessairement présentés. La mise en tampon des données qui défilent à l'extérieur de la plage d'affichage ou du champ actif ne doit pas être exigée (voir 8.3.3(11)).

Vidéotex: Exécution de tous les codes de commande du jeu de code C1 défini dans la norme.

Télétexte: Tous les codes de commande du jeu C1 doivent être exécutés sauf les codes ci-dessous dont l'exécution est liée à la matérialisation.

5/0 AVEC PROTECTION  
5/15 SANS PROTECTION

1) Champs sans protection

Télétexte: Ne s'applique pas. L'exécution des commandes AVEC PROTECTION et SANS PROTECTION doit rester sans effet.

Vidéotex: Doit comporter 40 champs sans protection, compte tenu des limitations de mémoire.

Les caractères du répertoire de caractères graphiques, du jeu mosaïque (y compris les codes mosaïques distincts), et du jeu JCDR seulement peuvent être mémorisés aux fins d'édition et de transmission subséquente (voir la note 1). La mémorisation n'est garantie que si ces caractères comportent une taille uniforme identique à celle qui a été établie par les dimensions du texte lorsque le champ était sans protection, ainsi que les attributs de texte par défaut pour la rotation, le trajet de caractère, l'espacement des caractères, l'espacement des rangées et la réflexion. Cette taille uniforme des caractères définit une grille qui est alignée avec l'origine du champ. La mémorisation des caractères est garantie seulement lorsque les champs de caractères respectifs s'alignent avec cette grille (voir la note 2). Lorsque l'alignement n'est pas obtenu, la mémorisation et l'affichage de ces caractères sont liés à la matérialisation, c'est-à-dire que les caractères peuvent être déplacés pour s'aligner puis mémorisés, ou

---



---

n'être ni déplacés ni mémorisés. De plus, il est seulement garanti que les caractères reçus de l'ordinateur central seront mémorisés si le champ actif coïncide avec le champ sans protection qui fait l'objet de l'écriture par l'ordinateur central. Les caractères qui se trouvent dans un champ sans protection, sans satisfaire à la condition ci-dessus concernant les attributs, peuvent être traités d'une manière liée à la matérialisation.

Toutes les autres données de la couche présentation sont affichées selon le processus d'affichage normal de la couche présentation sans être mémorisées.

La couleur peut changer d'un champ de caractère à l'autre dans un champ sans protection. Le mode de couleur peut changer d'un champ de caractère à l'autre dans un champ sans protection.

Les couleurs mémorisées pour l'attribut de couleur du champ sans protection sont les couleurs qui apparaissent dans la carte des couleurs et qui sont montrées à l'utilisateur. La carte des couleurs elle-même est seulement retransmise à l'ordinateur central indirectement lorsque le mode de couleur 0 est utilisé pour la transmission.

Il n'est pas nécessaire de mémoriser ni de transmettre subséquemment les définitions des configurations JCDR dans le cas d'un champ sans protection. Ces définitions peuvent ne pas être interprétées si elles sont transmises à un ordinateur central. Il est seulement nécessaire de transmettre (voir la note 3) les champs sans protection qui ont été édités par interaction (au clavier) de l'utilisateur.

---

---

Les données peuvent être introduites ou éditées dans des champs sans protection par interaction de l'utilisateur seulement si le curseur d'affichage est présent. L'introduction de données par l'utilisateur dans des champs sans protection ne doit pas entrer en conflit avec la réception et l'interprétation subséquente du code reçu de la couche présentation. Il est permis, pour la transmission des données dans des champs sans protection (par interaction de l'utilisateur), de supprimer le curseur d'affichage. Le contenu du champ sans protection devrait être maintenu jusqu'à ce que le champ sans protection soit protégé. Lorsque aucun champ sans protection n'est défini, ou lorsque le curseur d'affichage est supprimé, il existe un mode de communication lié à la matérialisation.

Le format de transmission est illustré au tableau D2 (voir les notes 1 et 4).

**Notes:**

(1) La transmission des données de l'appareil récepteur à l'ordinateur central est un processus distinct de la couche présentation indépendant du processus de présentation de l'ordinateur central à l'appareil récepteur. Le tableau D2 définit le code de la couche présentation que le terminal peut utiliser pour transmettre les données mémorisées dans un champ sans protection et le maximum de données qu'un ordinateur central doit interpréter. Si la transmission comporte des codes supplémentaires dépassant les exigences du MRS, un ordinateur central peut ne pas en tenir compte. Un terminal ne devrait pas transmettre des données supplémentaires entraînant la confusion de tout ordinateur central qui interprète les données seulement jusqu'à la limite MRS.

(2) L'alignement de la taille et de la position des caractères dans un champ sans protection ne peut pas être garanti si différentes longueurs d'opérande à valeurs multiples sont utilisées pour spécifier le champ et le caractère. Si les dimensions spécifiées correspondent à TEXTE NORMAL, TEXTE PETIT, DOUBLE HAUTEUR ou DOUBLE TAILLE, l'alignement de la position est garanti seulement si les caractères de commande de mise en page sont utilisés pour positionner les caractères relativement à l'origine du champ.

---

---

(3) Lorsque l'utilisateur amorce la transmission d'un champ sans protection, au moins l'information nécessaire à chaque transmission devrait être transmise (voir tableau D2), même si aucun champ sans protection n'est transmis.

(4) Le tableau D2 indique le format de transmission de l'information relative à un champ sans protection. Comme l'indique la colonne "Condition", certains codes sont facultatifs et ne sont requis que s'il est nécessaire d'assurer qu'ils se conforment à la syntaxe de données et qu'ils représentent avec précision l'image mémorisée dans le champ sans protection (compte tenu des contraintes déjà décrites). Différentes applications peuvent comporter différentes exigences d'entrée. Il est souhaitable que les terminaux tentent de minimiser le nombre de codes transmis, et que chaque champ soit indépendant du contexte des autres champs. Plus particulièrement, le rétablissement des désignations et des appels des jeux G par défaut dans chaque champ est spécialement important pour l'indépendance à l'égard du contexte.

(5) La spécification des coordonnées de l'origine du champ pour la transmission doit être identique à celle de la définition du champ.

---

Tableau D2  
Format de transmission des champs sans protection

Occurrence	Code	Condition (voir note 4)
Par transmission de bloc de champ sans protection	NSR	Nécessaire au commencement d'une transmission. Facultatif avant les autres champs.
	DOMAINE	Nécessaire seulement si le code est différent du code par défaut.
Par champ sans protection	CHAMP (voir note 5)	Nécessaire pour chaque champ
	TEXTE	Nécessaire seulement si les dimensions du texte sont différentes des dimensions par défaut ou de celles qui ont été spécifiées dans le dernier champ.
	DÉTERMINATION POINT (ou caractère de commande de mise en page)	Nécessaire seulement pour localiser le premier caractère du champ, s'il n'est pas déjà localisé.
Par caractère dans un champ sans protection	SÉLECTION COULEUR (ou DÉTERMINATION COULEUR en mode de couleur 0)	Nécessaire seulement si la couleur change.
	Codes de caractère (y compris les séquences d'extension de code appropriées) et tout caractère de commande de mise en page, ESPACE, RÉPÉTITION, RÉPÉTITION JUSQU'À EOL	Nécessaire seulement pour positionner les caractères dans le champ.
	FIN	Nécessaire à la fin de chaque champ.
	DÉTERMINATION POINT	Nécessaire pour établir, à l'ordinateur central, la position de la prochaine introduction par l'utilisateur.
	SDC	Nécessaire pour délimiter logiquement la fin de transmission du champ sans protection.

