

## RECOMMANDATION UIT-R BS.1196\*

**CODAGE AUDIO POUR LA RADIODIFFUSION DE TÉLÉVISION  
NUMÉRIQUE PAR VOIE DE TERRE**

(Questions UIT-R 78/10, UIT-R 208/10, UIT-R 211/10 et UIT-R 121/11)

(1995)

L'Assemblée des radiocommunications de l'UIT,

*considérant*

- a) que la radiodiffusion de télévision numérique par voie hertzienne de Terre se fera dans les bandes des ondes métriques et décimétriques;
- b) qu'il est essentiel d'utiliser pour la partie son, un système audio multivoie de haute qualité avec une réduction du débit binaire efficace;
- c) que les systèmes de codage du son avec réduction du débit binaire doivent être protégés contre les erreurs binaires résiduelles résultant du décodage et du démultiplexage du canal;
- d) que les systèmes audio multivoies avec ou sans image associée font l'objet de la Recommandation UIT-R BS.775;
- e) que l'évaluation subjective des dégradations faibles dans les systèmes audio, y compris les systèmes audio multivoies, fait l'objet de la Recommandation UIT-R BS.1116;
- f) que l'usage de techniques de codage de source audio similaires pour différents services pourrait permettre une plus grande souplesse d'utilisation des systèmes et une diminution du coût des récepteurs;
- g) que la radiodiffusion sonore numérique à destination de récepteurs fixes, portatifs ou placés à bord de véhicules par voie hertzienne de Terre dans les bandes des ondes métriques et décimétriques fait l'objet de la Recommandation UIT-R BS.774 et UIT-R BS.1114;
- h) que des systèmes génériques de réduction du débit binaire audio ont été étudiés à l'ISO/CEI en liaison avec l'UIT-R et que ces études ont permis l'élaboration de la Norme internationale IS 11172-3 (Codage MPEG-1 audio) et de la Norme internationale IS 13818-3 (Codage MPEG-2 audio), et font l'objet de la Recommandation UIT-R BS.1115;
- j) que plusieurs exploitants de services de radiodiffusion sonore par satellite et de nombreux systèmes de distribution secondaire (télévision par câble) utilisent ou ont indiqué qu'ils prévoient d'appliquer pour leurs services numériques le codage audio MPEG-1, ou le codage audio multivoie MPEG-2 audio ou AC-3 (voir les Annexes);
- k) que de nombreux équipements sont conformes aux Normes internationales IS 11172-3 (MPEG-1 audio) et 13818-3 (MPEG-2 audio);
- l) qu'un important système audionumérique cinématographique utilise le codage AC-3;
- m) que le système européen de télévision numérique (DVB) utilisera le codage MPEG-2 audio;
- n) que le système nord-américain de télévision numérique évoluée (ATV) utilisera le codage AC-3;
- o) que l'interopérabilité avec d'autres supports tel le disque optique utilisant le codage audio MPEG-2 ou AC-3 est importante,

*recommande*

**1** d'utiliser, pour le codage audio dans les systèmes de radiodiffusion télévisuelle par voie hertzienne de Terre, la norme internationale de codage audio spécifiée dans l'Annexe 1 ou la norme nationale des Etats-Unis d'Amérique spécifiée dans l'Annexe 2.

---

\* Cette Recommandation doit être portée à l'attention de l'Organisation internationale de standardisation (ISO) et de la Commission électrotechnique internationale (CEI).

NOTE 1 – Il convient de noter que les débits binaires audio nécessaires pour obtenir les niveaux de qualité spécifiés pour le son multivoie avec ces systèmes n'ont pas été totalement évalués et documentés au sein de l'UIT-R.

NOTE 2 – Il convient aussi de noter que des améliorations compatibles sont en cours de développement (exemple: utilisation plus poussée des propriétés syntactiques et modélisation psycho-acoustique améliorée), qui sont susceptibles avec le temps d'améliorer de façon significative les performances des systèmes.

NOTE 3 – Reconnaissant que l'évaluation des performances actuelles et futures de ces systèmes de codage intéresse principalement la Commission d'études 10 des radiocommunications, celle-ci est invitée à poursuivre d'urgence ses travaux dans ce domaine avec pour objectif de pouvoir apporter des éléments fiables à ajouter à la présente Recommandation et détailler les caractéristiques de qualité de fonctionnement des options de codage disponibles.

NOTE 4 – Le système de codage audio spécifié dans l'Annexe 2 est un codec non rétrocompatible (NBC) et n'est donc pas compatible avec le codage à deux voies conforme à la Recommandation UIT-R BS.1115.

NOTE 5 – Les Commissions d'études 10 et 11 des radiocommunications sont encouragées à poursuivre leurs efforts pour parvenir à l'unification des spécifications de codage.

## ANNEXE 1

### **Couche audio II de la norme MPEG (ISO/CEI IS 13818-3): codage générique bicanal ou multicanal du son pour radiodiffusion vidéonumérique, radiodiffusion audionumérique et multimédia informatique**

#### **1 Introduction**

De 1988 à 1992, l'Organisation internationale de normalisation (ISO) a élaboré et préparé une norme intitulée Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s. Le «Sous-groupe Audio» du Groupe d'experts en images animées (MPEG) avait la responsabilité de mettre au point une norme pour le codage générique des signaux audio MIC avec des fréquences d'échantillonnage de 32, 44,1 et 48 kHz, à des débits binaires compris entre 32 et 192 kbit/s par canal monophonique et entre 64 et 384 kbit/s par canal stéréophonique. Ces travaux ont abouti à la partie audio de la norme MPEG-1, qui se compose de trois couches ayant des complexités différentes selon les applications et qui porte le numéro ISO/CEI 11172-3 [1]. A l'issue d'essais intensifs en 1992 et en 1993, l'UIT-R recommande d'utiliser la couche II de la norme MPEG-1 pour la contribution, la distribution et l'émission, qui sont des applications typiques de radiodiffusion [2]. En ce qui concerne les applications de télécommunication, l'UIT-T a établi la Recommandation UIT-T J.52 [3] qui est la norme relative à la transmission sur RNIS de données audio à codage MPEG.

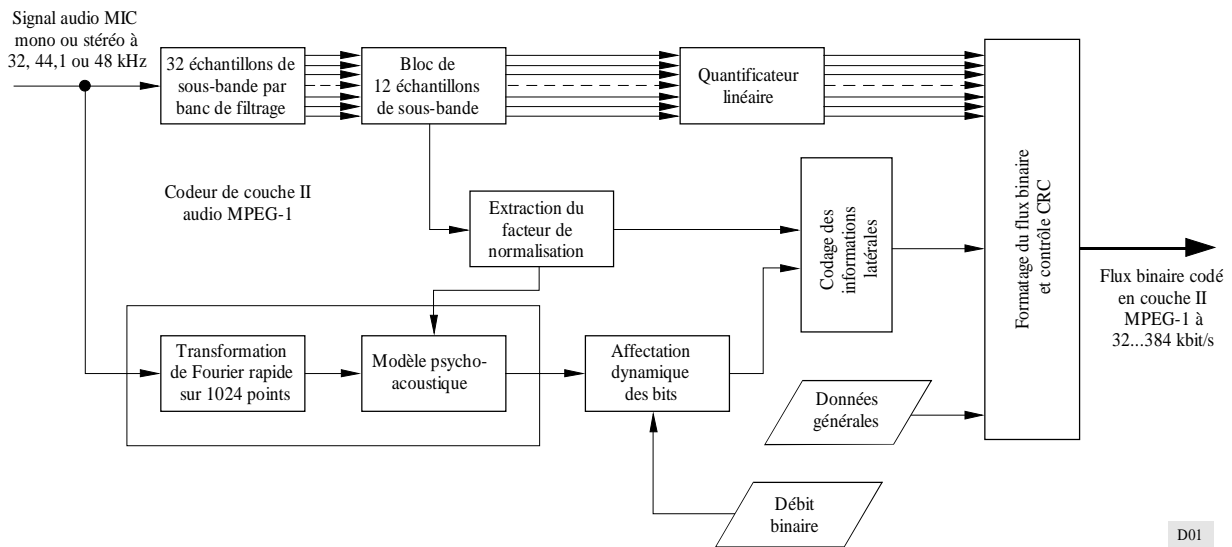
Le premier objectif du Sous-groupe Audio MPEG-2 était l'extension de deux à cinq canaux du codage audio de haute qualité, de façon à conserver la rétrocompatibilité et à respecter les Recommandations UIT-R, de la SMPTE (Society of Motion Picture and Television Engineers) et de l'Union européenne de radio-télévision (UER). Cette extension a été réalisée en novembre 1994 avec l'approbation de la norme ISO/CEI 13818-3, dite Audio MPEG-2 [4]. Cette norme décrit le codage en haute qualité de 5,1 canaux audio, c'est-à-dire cinq canaux de pleine largeur de bande plus un canal à bande étroite pour le renforcement des basses fréquences. Ce codage est compatible, en amont, avec la norme MPEG-1, ce qui est la garantie que les décodeurs bicanaux existants resteront capables de décoder les informations stéréophoniques compatibles à partir de signaux multicanaux. Pour la reproduction des sons périphériques, on utilise, conformément à la norme 3/2, les positions suivantes du haut-parleur: à gauche, au centre et à droite plus gauche et droite périphériques. Les applications envisagées concernent les systèmes numériques de télévision tels que dTTb (diffusion terrienne de la TV numérique), TVHD-T, HD-SAT, ADTT, supports d'enregistrement numérique comme le vidéodisque et le système de radiodiffusion audionumérique (RAN) (EU147) (Recommandation UIT-R BS.1114).

Le deuxième objectif du Sous-groupe Audio MPEG-2 était l'extension des signaux audio MPEG-1 vers les très faibles fréquences d'échantillonnage afin d'améliorer la qualité du son aux débits binaires inférieurs à 64 kbit/s par canal, en particulier pour les applications vocales. Cette extension présente un intérêt particulier pour les applications RNIS en bande étroite dans lesquelles, pour de simples raisons d'exploitation, on peut éviter de multiplexer plusieurs canaux B tout en continuant à assurer une excellente qualité audio, même avec des débits binaires descendant jusqu'à 48 kbit/s. Une autre application importante est le système de radiodiffusion audionumérique RAN EU147. La capacité en programmes du principal canal de service peut être augmentée si l'on applique l'option de basse fréquence d'échantillonnage à des canaux d'actualités de haute qualité nécessitant moins d'éléments binaires, à qualité égale, qu'à la pleine fréquence d'échantillonnage.

## 2 Principes de la technique de codage des signaux audio MPEG de couche II

On peut faire appel à deux procédés pour diminuer le débit binaire des signaux audio. Le premier consiste essentiellement à supprimer l'inapplicabilité du signal audio en utilisant des corrélations statistiques. Par ailleurs, cette nouvelle génération de systèmes de codage diminue la redondance du signal audio en tenant compte de lois psycho-acoustiques comme le masquage spectral et le masquage temporel. Ce n'est qu'au moyen de ces deux techniques, faisant appel aux corrélations statistiques et aux effets de masquage de l'oreille humaine, que l'on a pu obtenir une notable diminution du débit, jusqu'à 200 kbit/s par signal stéréophonique et moins.

FIGURE 1  
Schéma fonctionnel du codeur de couche II selon l'ISO/CEI 11172-3  
(audio MPEG-1)



D01

La couche II correspond au système de codage audio bien connu MUSICAM (codage et multiplexage en sous-bandes, adaptés à la structure du masquage), tandis que la couche I est à interpréter comme une version simplifiée du système MUSICAM. La structure fondamentale du procédé de codage (qui est plus ou moins commune aux deux couches, I et II) se caractérise par le fait que les signaux audio MPEG utilisent un codage de la perception acoustique. Le codeur se compose donc des principaux modules suivants:

- Une des principales fonctions du codeur consiste à convertir les signaux MIC d'entrée à 20 kHz de largeur dans le domaine temporel en raies spectrales sous-échantillonnées. Pour cela, on utilise dans les deux couches un banc de filtrage polyphasé qui se compose de 32 sous-bandes équidistantes.
- Pour calculer une estimation du seuil de masquage réel, qui dépend du temps, on applique au signal audio MIC large bande, parallèlement au filtrage, la sortie d'une transformation de Fourier. A cette fin, on fait appel à un modèle de perception psycho-acoustique fondé sur les règles connues en la matière, en tant que bloc fonctionnel additionnel dans le codeur. Ce bloc simule le masquage spectral et, dans une certaine mesure, le masquage temporel également. La base principale pour le calcul du seuil de masquage dans le codeur est donnée par les résultats de mesurages du seuil de masquage effectués sur des signaux à bande étroite compte tenu du bruit de masquage des tonalités et inversement. Concernant l'excursion de fréquence et la différence de niveau de pression acoustique, on ne trouve dans la littérature que des relations très limitées et artificielles entre composantes de masquage et tonalités d'essai. Les résultats les moins favorables en ce qui concerne les pentes supérieure et inférieure des courbes de masquage ont été pris en compte pour construire l'hypothèse que l'on peut utiliser les mêmes seuils de masquage pour les deux situations: audio simple et audio complexe.
- Les échantillons de sous-bande sont quantifiés et codés dans l'intention de maintenir le bruit (introduit par la quantification) au-dessous du seuil de masquage. Les couches I et II utilisent une technique d'expansion de bloc (d'une capacité de 12 échantillons de sous-bande) mettant en œuvre un facteur de normalisation codé sur 6 bits et applicable à une dynamique d'environ 120 dB. Grâce à cette technique de normalisation, les couches I et II peuvent traiter une dynamique beaucoup plus étendue que les disques compacts ou le système DAT, qui utilisent des mots MIC de 16 bits conventionnels.

- Dans le cas des signaux stéréophoniques, on peut ajouter le codage stéréophonique mixte en tant que caractéristique supplémentaire permettant d'exploiter la redondance et l'inapplicabilité des données de programme stéréophonique normales. Ce procédé peut servir à augmenter la qualité audio aux faibles débits binaires et/ou à réduire le débit binaire pour des signaux stéréophoniques. L'augmentation de complexité du codeur et, indirectement, requise du décodeur, est négligeable. Il importe de signaler que le codage stéréophonique mixte n'augmente pas le délai de codage global.
- Après codage du signal audio, on utilise un macrobloc pour mettre en trames audio consécutives le flux binaire audio MPEG. Dans la couche I, la longueur d'une trame correspond à 384 échantillons audio à codage MIC. Dans la couche II, la longueur d'une trame correspond à 1 152 échantillons audio MIC. Chaque trame audio commence, comme indiqué sur la Fig. 2, par un en-tête suivi de l'information d'affectation des bits, du facteur de normalisation et des échantillons de sous-bande, quantifiés et codés. Chaque trame audio se termine par le champ dit de données auxiliaires, dont la longueur est variable et qui peut être spécifiée pour certaines applications.

## 2.1 Modèle psycho-acoustique

Le modèle psycho-acoustique calcule le seuil de masquage minimal qui est nécessaire pour déterminer le seuil de perception du niveau de bruit pour chaque bande du banc de filtrage. La différence entre le niveau maximal du signal et la valeur minimale du seuil de masquage est utilisée lors de l'affectation des bits ou du bruit pour déterminer le niveau réel du quantificateur de chaque bloc dans chaque sous-bande. La partie informative de l'ISO/CEI 11172-3 présente deux modèles psycho-acoustiques, qui peuvent être tous les deux appliqués à une couche quelconque de l'algorithme audio du MPEG. Mais en pratique, seul le modèle 1 sera utilisé pour les couches I et II et le modèle 2 pour la couche III. Dans les deux modèles psycho-acoustiques, la sortie finale du modèle correspond à un niveau du rapport signal sur masquage pour chaque sous-bande de la couche II. Un modèle psycho-acoustique n'est nécessaire que dans le codeur, ce qui permet de conserver des décodeurs notablement moins complexes. Il est donc possible d'améliorer ultérieurement la performance du codeur, en terme de rapport entre débit binaire et qualité subjective. Pour certaines applications qui n'exigent pas un très faible débit binaire, il est même possible de faire appel à un codeur très simple, sans aucun modèle psycho-acoustique.

Pour un calcul adéquat des seuils de masquage dans le domaine fréquentiel, il convient de partir d'une haute résolution spectrale c'est-à-dire de sous-bandes étroites dans la région des basses fréquences, avec en revanche une résolution plus faible dans la région des hautes fréquences, avec des sous-bandes plus larges. Cela se traduira par une structure arborescente du banc de filtrage. Le réseau de filtrage polyphasé qui est utilisé pour le filtrage en sous-bande possède une structure parallèle qui ne découpe pas les sous-bandes en différentes largeurs. Un important avantage du banc de filtrage est cependant qu'il optimise l'adaptation des blocs audio aux nécessités des effets de masquage temporel et des prééchos inaudibles. Le deuxième avantage important de ce banc est qu'il n'apporte que peu de délai et de complexité. Pour compenser le manque de précision dans l'analyse spectrale du banc de filtrage, on utilise une transformation de Fourier rapide (TFR) de 1 024 points dans la couche II en parallèle au filtrage du signal audio dans 32 sous-bandes. On utilise la sortie de la TFR afin de déterminer les composantes de masquage tonales (sinusoïdaux) et atonales (parasites) du signal audio proprement dit. La recherche en psycho-acoustique a bien montré que la hauteur tonale d'une composante de masquage avait une influence sur le seuil de masquage. C'est pourquoi il est utile de différencier les composantes tonales et atonales. Les seuils de masquage individuels de chaque composante de masquage, situés au-dessus du seuil de masquage absolu, sont calculés en fonction de la position en fréquence, du niveau de sonie et de la tonie. Chaque seuil de masquage, y compris le seuil absolu, est individuellement ajouté au seuil dit *de masquage global*. Dans chaque sous-bande, on détermine la valeur minimale de cette caractéristique de masquage. Finalement, on calcule pour chaque sous-bande et pour chaque bloc la différence entre le niveau maximal du signal (calculé aussi bien par les facteurs de normalisation que par la densité spectrale de puissance de la TFR) et le seuil minimal de masquage. La longueur des blocs de la couche II est déterminée par 36 échantillons de sous-bande, correspondant à 1 152 échantillons MIC d'entrée audio. Cette différence (entre le niveau maximal du signal et le seuil de masquage minimal) est appelée rapport signal sur masquage et constitue la fonction d'entrée applicable à l'affectation des bits.

La Fig. 1 présente le codeur de couche II sous forme de diagramme fonctionnel. Les paragraphes suivants expliquent en détail chaque étape des processus de codage et de décodage, y compris la subdivision du signal audio MIC d'entrée en 32 sous-bandes équidistantes au moyen d'un banc de filtrage à analyse polyphasée; une fonction d'affectation dynamique des bits selon les règles d'un modèle psycho-acoustique, la technique de compression-expansion des blocs des échantillons de sous-bande et la mise en forme du flux binaire.

## 2.2 Banc de filtrage

Le prototype du filtre miroir en quadrature est d'ordre 511, optimisé en termes de résolution spectrale, avec un niveau de suppression des lobes secondaires supérieur à 96 dB. Cette suppression est nécessaire pour un affaiblissement suffisant des distorsions de repliement. Ce banc de filtrage assure un compromis raisonnable entre le comportement temporel d'un côté et la résolution spectrale de l'autre. Une fonction de mise en correspondance temps/fréquence fournissant un grand nombre de sous-bandes facilite la réduction du débit binaire. Cela est dû au fait que l'oreille

humaine perçoit les informations audio avec une résolution dans le domaine spectral qui correspond aux bandes critiques de l'oreille, ou même au-dessous de ces bandes. Celles-ci ont une largeur d'environ 100 Hz dans la région des basses fréquences, c'est-à-dire au-dessous de 500 Hz, et une largeur d'environ 20% de la fréquence centrale aux fréquences supérieures. L'exigence d'une bonne résolution spectrale s'oppose malheureusement à la nécessité de maintenir entre certaines limites (par rapport à l'attaque d'un son de percussion) la réponse aux impulsions transitoires (appelée préécho et post-écho), en termes de position temporelle et d'amplitude. La connaissance de l'allure de la courbe de masquage temporel renseigne sur la position temporelle et sur l'amplitude nécessaires du préécho produit par le découpage temps/fréquence, de manière que ce préécho (normalement beaucoup plus critique que le post-écho) soit masqué par l'attaque percussive originale. Associée au banc de filtrage à double synthèse qui est situé dans le décodeur, cette technique de filtrage donne une fonction de transfert global qui est optimisée en terme de perception d'une réponse à une impulsion parfaite.

Dans le décodeur, le banc de filtrage à double synthèse reconstruit un bloc de 32 échantillons de sortie. La structure du filtre est très efficace pour mettre en œuvre un décodeur de faible complexité et sans traitement du signal, car il exige généralement moins de 80 opérations de multiplication/addition sur des entiers, par échantillon MIC de sortie. En outre, le filtre d'analyse et de synthèse complet n'apporte qu'un délai global de 10,5 ms à la fréquence d'échantillonnage de 48 kHz.

### 2.3 Détermination et codage des facteurs de normalisation

Le calcul du facteur de normalisation pour chaque sous-bande est effectué sur un bloc de 12 échantillons de sous-bande. Le maximum de la valeur absolue de ces 12 échantillons est déterminé et quantifié en mots d'une longueur de 6 bits, couvrant une dynamique totale de 120 dB par sous-bande avec une résolution de 2 dB par classe de facteurs de normalisation. Dans la couche I, on transmet un facteur de normalisation pour chaque bloc et pour chaque sous-bande sans affectation de bit zéro.

La couche II fait appel à un codage additionnel afin de diminuer la vitesse de transmission des facteurs de normalisation. Etant donné que, dans la couche II, une trame correspond à 36 échantillons de sous-bande, c'est-à-dire à trois longueurs de trame de couche I, trois facteurs de normalisation doivent en principe être transmis. Pour réduire le débit binaire des facteurs de normalisation, on a étudié une stratégie de codage qui exploite les effets de masquage temporel de l'oreille. Trois facteurs de normalisation successifs, de chaque sous-bande d'une même trame, sont regroupés et rangés dans certains motifs de facteur de normalisation. Selon le motif, on transmet un, deux ou trois facteurs de normalisation de concert avec une information additionnelle pour la sélection du facteur de normalisation, composée de 2 bits par sous-bande. S'il n'y a que de petits écarts entre un facteur et le suivant, seul le plus grand est à transmettre. Cela se produit relativement souvent pour des sons de hauteur tonale constante. Si des attaques de sons percussifs doivent être codées, il faut transmettre deux des facteurs de normalisation ou les trois, selon la pente de montée et de descente de l'attaque. Cette technique de codage additionnelle permet en moyenne une réduction de moitié du débit des facteurs de normalisation par rapport à la couche I.

### 2.4 Affectation des bits et codage des informations correspondantes

Avant le réglage sur un débit binaire fixe, il faut déterminer le nombre de bits disponibles pour le codage des échantillons. Ce nombre dépend de celui des bits requis pour les facteurs de normalisation, pour les informations de sélection de facteur de normalisation, pour les informations d'affectation des bits et pour les données auxiliaires.

On détermine la procédure d'affectation des bits en minimisant le rapport bruit total sur masquage dans chaque sous-bande et dans la trame complète. Cette procédure est de nature récurrente: à chaque étape d'itération, on augmente le nombre de niveaux de quantification de la sous-bande qui présente le meilleur résultat, avec la contrainte que le nombre de bits utilisés ne doit pas dépasser celui qui est disponible pour cette trame. La couche II utilise 4 bits pour le codage des informations d'affectation des bits pour les plus basses sous-bandes et seulement 2 bits pour les plus hautes, par trame audio.

### 2.5 Quantification et codage des échantillons de sous-bande

On normalise d'abord chacun des 12 échantillons de sous-bande d'un même bloc en divisant sa valeur par le facteur de normalisation. Le résultat est quantifié en fonction du nombre de bits requis par le bloc d'affectation des bits. Seuls des nombres impairs de niveaux de quantification sont possibles, permettant une représentation exacte d'un zéro numérique. La couche I utilise 14 classes de quantification différentes, contenant  $2^n - 1$  échelons, avec  $2 \leq n \leq 15$  niveaux de quantification différents. Il en va de même pour toutes les sous-bandes. Par ailleurs, si aucun bit n'est affecté à une sous-bande, aucune quantification que ce soit ne peut être utilisée.

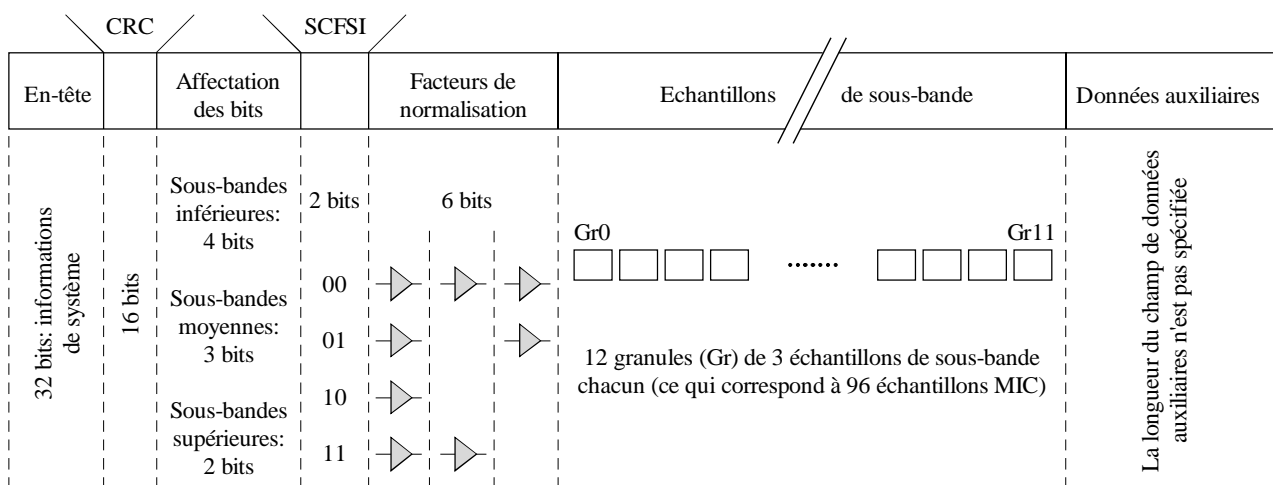
Dans la couche II, le nombre de niveaux de quantification différents dépend du nombre de sous-bandes mais l'étendue des niveaux de quantification ira toujours de 3 à 65 535 avec la possibilité supplémentaire d'absence de quantification. On peut quantifier les échantillons des sous-bandes dans la région des basses fréquences avec 15 niveaux de quantification, 7 dans la région des moyennes fréquences et 3 niveaux seulement dans la région des hautes fréquences. Les classes de quantification peuvent contenir 3, 5, 7, 9, 15, 63, ..., 65 535 niveaux de quantification. Etant donné que les niveaux de quantification aux nombres de 3, 5 et 9 ne permettent pas d'utiliser efficacement un mot de code composé de 2, 3 ou 4 bits seulement, on regroupe trois échantillons successifs de sous-bande pour former un «granule». Celui-ci est ensuite codé en un seul mot. Le gain de codage obtenu par ce groupage atteint 37,5%. Etant donné que de nombreuses sous-bandes, surtout dans la région des hautes fréquences, sont normalement quantifiées avec seulement 3, 5, 7 et 9 niveaux de quantification, le facteur de réduction de la longueur des mots de code est digne d'intérêt.

## 2.6 Structure du flux binaire de couche II

Le flux binaire de couche II a été construit de telle manière que l'on puisse utiliser aussi bien un décodeur de faible complexité qu'un faible délai de décodage. Il est également tel que le signal audio codé contienne beaucoup de points d'entrée correspondant à des créneaux temporels étroits et constants. La représentation numérique codée d'un algorithme de codage efficace, spécialement adapté aux applications de stockage, doit autoriser des multiples des points d'entrée dans le flux de données codées afin d'enregistrer, de lire et de corriger de brèves séquences audio et afin de définir précisément les positions de correction. Pour permettre une mise en œuvre simple du décodeur, la trame qui est comprise entre ces points d'entrée doit toujours contenir l'ensemble des informations nécessaires pour décoder le flux binaire. En plus des différentes applications, une telle trame doit acheminer toutes les informations nécessaires pour autoriser une grande étendue de codage avec un grand nombre de paramètres différents. Ces propriétés sont importantes également dans le domaine de la radiodiffusion sonore numérique, où un décodeur de faible complexité est nécessaire pour des raisons d'ordre économique et où de fréquents points d'entrée dans le flux binaire sont nécessaires pour permettre un masquage aisé de blocs d'échantillons consécutifs, dégradés par des erreurs sporadiques.

La Fig. 2 montre le format du flux binaire audio codé en couche II. La structure du flux binaire est caractérisée par de brèves trames audio autonomes, correspondant à 1 152 échantillons à codage MIC. Chaque trame qui commence par un mot de synchronisation de 12 bits peut être localisée et décodés individuellement. Sa durée est de 24 ms à la fréquence d'échantillonnage de 48 kHz.

FIGURE 2  
Structure d'une trame audio dans le flux binaire codé en couche II  
selon l'ISO/CEI 11172-3



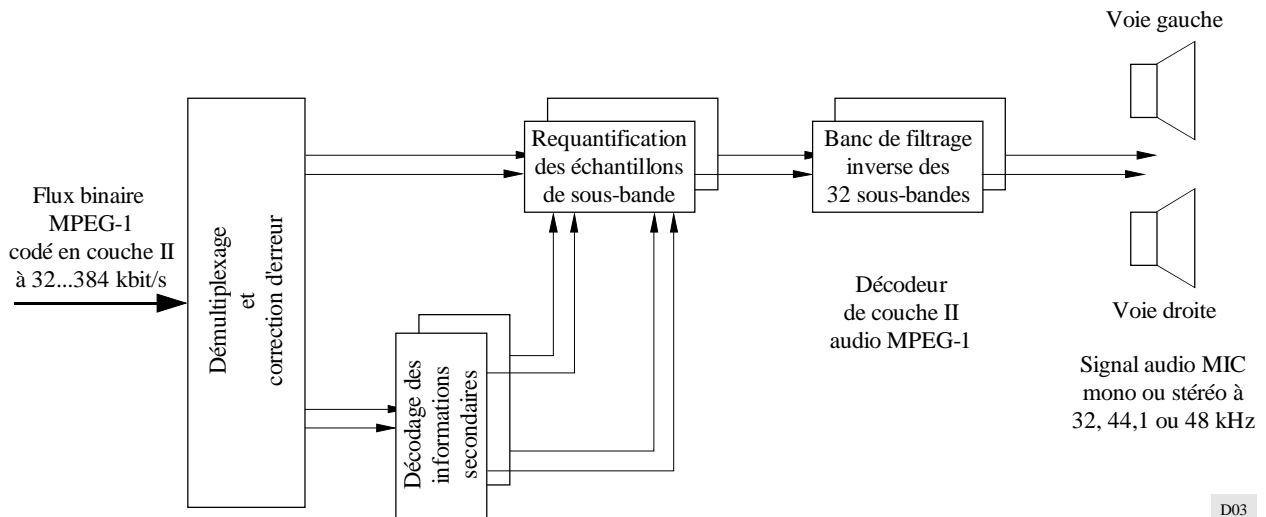
La trame est assemblée sur la base de 1 152 échantillons audio en codage MIC.  
A la fréquence d'échantillonnage de 48 kHz, une telle trame a donc une durée de 24 ms.

## 2.7 Décodage de la couche II

Le schéma fonctionnel du décodeur est représenté sur la Fig. 3. Tout d'abord, on sépare du flux binaire de couche II selon l'ISO/MPEG/AUDIO les informations d'en-tête, le CRC, les informations secondaires c'est-à-dire l'information d'affectation des bits avec les facteurs de normalisation, et 12 échantillons successifs de chaque signal de sous-bande.

FIGURE 3

Schéma fonctionnel du décodeur de couche II (audio MPEG-1)  
selon l'ISO/CEI 11172-3



D03

Le processus de reconstruction pour obtenir de nouveau le signal audio MIC est caractérisé par le remplissage du format de données des échantillons de sous-bande en termes de facteur de normalisation et d'affectation des bits, pour chaque sous-trame et chaque trame. Le banc de filtrage à double synthèse reconstitue le signal audio complet en bande large, pouvant atteindre 24 kHz. Le processus de décodage nécessite nettement moins de puissance de calcul que le processus de codage. Pour la couche II, le rapport est de 1/3. Etant donné la faible puissance de calcul nécessaire et la structure non complexe de l'algorithme, la couche II a pu être facilement réalisée dans des circuits VLSI particuliers. Depuis 1993, plusieurs constructeurs proposent des puces de décodage stéréophonique. On trouve des codeurs stéréophoniques de couches I et II qui sont réalisés en un seul processeur de signaux à virgule fixe (DSP56002).

## 3 Signaux audio MPEG-2: codage générique multicanal audio

Une des caractéristiques principales de la norme audio MPEG-2 (ISO/CEI 13818-3) est la rétrocompatibilité avec les programmes audio codés selon l'ISO/CEI 11172-3 en monophonie, en stéréophonie ou en deux canaux distincts. C'est-à-dire qu'un décodeur audio selon l'ISO/CEI 11172-3 ou la norme MPEG-1 est en mesure de décoder correctement les informations stéréophoniques de base d'un programme multicanal. Les informations stéréophoniques de base sont conservées dans les canaux de gauche et de droite qui contiennent un assortiment réduit mais approprié des informations audio contenues dans tous les canaux.

La rétrocompatibilité avec la stéréophonie à deux voies est une exigence impérative pour de nombreux fournisseurs de service qui pourront proposer, ultérieurement, des signaux audionumériques de haute qualité avec voies périphériques. A l'exception de l'univers du cinéma, il n'existe actuellement pas de composants audionumériques multicanaux. Les puces de décodage MPEG-1 en couches I et II sont toutefois largement répandues. Elles supportent les signaux monophoniques et stéréophoniques. En raison de la rétrocompatibilité de la norme de codage multicanal audio MPEG, un tel décodeur bicanal fournira toujours un signal stéréophonique correct avec toutes les informations audio extraites du flux binaire multicanal audio MPEG-2.

La norme audio MPEG-1 a été étendue, dans le cadre de l'activité MPEG-2 en direction des très basses fréquences d'échantillonnage, afin d'améliorer la qualité audio des signaux monophoniques et stéréophoniques conventionnels pour des débits égaux ou inférieurs à 64 kbit/s par canal, en particulier pour les applications de commentaire. Ce but a été atteint par une réduction de la fréquence d'échantillonnage à 16, 22,05 ou 24 kHz, correspondant à une largeur de bande inférieure ou égale à 7,5, 10,5 ou 11,5 kHz. La seule différence par rapport à la norme MPEG-1 est une modification des tables de débits binaires et d'affectation des bits dans le codeur et dans le décodeur. Les principes de codage et de décodage mis en œuvre dans les couches audio MPEG-1 sont totalement conservés.

### 3.1 Caractéristiques du système de codage multicanal audio MPEG-2

Il y a lieu que tout système générique à plusieurs canaux audionumériques, applicable à la télédiffusion et à l'enregistrement des images de télévision et du son, ainsi qu'à d'autres applications que la télédiffusion, réponde à plusieurs prescriptions de base et offre un certain nombre de spécificités techniques et opérationnelles. Etant donné qu'au cours des prochaines années la restitution stéréophonique normale va continuer à jouer un rôle dominant dans la plupart des applications grand public, la compatibilité avec le mode bicanal est une des exigences fondamentales. Il s'y ajoute l'interopérabilité entre médias différents, l'infracompatibilité avec des formats sonores se composant d'un plus petit nombre de canaux audio et donc une performance réduite en terme de son périphérique. Pour permettre des applications aussi universelles que possible, il importera également de tenir compte d'autres aspects, comme les services multilingues, le dialogue pur et la compression de dynamique.

Le codage audio MPEG-2 autorise une large gamme de débits binaires, de 32 à 1 066 kbit/s, cela grâce à une division en deux parties de la trame audio MPEG-2:

- le flux binaire primaire qui achemine les informations stéréophoniques compatibles avec la norme MPEG-1, au débit maximal de 384 kbit/s; et
- le flux binaire d'extension, qui achemine tout ou partie des informations spécifiquement MPEG-2, c'est-à-dire les données multicanaux et multilingues, non compatibles avec un décodeur audio à la norme MPEG-1.

Le flux binaire primaire atteint un débit maximal de 448 kbit/s dans la couche I et de 384 kbit/s dans la couche II. Le flux binaire d'extension réalise le débit complémentaire. Si, dans le cas de la couche II, un total de 384 kbit/s est choisi, le flux binaire d'extension peut être omis. Il n'est pas nécessaire que le débit soit fixe, car le codage MPEG-2 autorise le débit binaire variable, ce qui peut être intéressant en transmission par mode ATM ou dans des applications de stockage, comme le vidéodisque numérique (DVD).

Cette large gamme de débits binaires est compatible avec des applications qui exigent un faible débit binaire et une haute qualité audio, par exemple s'il n'y a qu'un seul processus de codage et qu'on peut éviter une mise en cascade. Elle est également compatible avec des applications dans lesquelles des débits binaires plus élevés, c'est-à-dire jusqu'à 180 kbit/s par canal, pourraient être souhaitables s'il faut tenir compte d'une mise en cascade ou d'un post-traitement. Des expériences conduites par un groupe d'experts de l'UIT-R ont montré qu'un processus de codage pouvait – sans dégradation subjective notable – être répété 9 fois en couche II de la norme MPEG-1, à condition que le débit soit assez élevé, c'est-à-dire 180 kbit/s par canal. Si le débit binaire est cependant de 120 kbit/s seulement, il convient de ne pas associer plus de 3 processus de codage.

#### 3.1.1 Performance en terme de présentation stéréophonique 3/2

Le système à 5 canaux recommandé par l'UIT-R, la SMPTE et l'UER est appelé stéréo 3/2 (3 voies frontales/2 voies périphériques). Il oblige à gérer cinq canaux dans la chaîne du studio, du support d'enregistrement, de la contribution, de la distribution, des liaisons d'émission jusqu'au domicile.

#### 3.1.2 Rétrocompatibilité et précompatibilité avec l'ISO/CEI 11172-3

Pour plusieurs applications, on a l'intention d'améliorer, peu à peu, le système stéréophonique de type 2/0 existant en émettant des voies sonores supplémentaires (centre, périphérie) sans faire appel à la diffusion de deux sources distinctes. Le décodeur de son multicanal doit être rétro/précompatible avec le format sonore existant.

Le terme *rétrocompatibilité* signifie que le décodeur à deux voies existant (à faible prix de revient) doit décoder correctement les informations stéréophoniques de base de type 2/0 contenues dans le flux binaire multicanal (voir la Fig. 4). Cela implique la mise en place de matrices d'analyse de compatibilité [5] faisant appel à des coefficients de réduction adéquats de l'assortiment afin de créer les signaux stéréophoniques compatibles,  $L_0$  et  $R_0$ , représentés sur la Fig. 5, qui montre également la matrice inverse qui permet (par dématricage), de récupérer, dans le décodeur MPEG-2, les cinq canaux audio distincts. Les équations de la matrice de base utilisée dans le codeur pour convertir les cinq signaux d'entrée,  $L$ ,  $R$ ,  $C$ ,  $L_s$  et  $R_s$ , en cinq canaux de transport,  $T_0$ ,  $T_1$ ,  $T_2$ ,  $T_3$  et  $T_4$ , sont les suivantes:

$$T_0 = L_0 = (\alpha \cdot L) + \beta (\alpha \cdot C) + \gamma (\alpha \cdot L_s)$$

$$T_1 = R_0 = (\alpha \cdot R) + \beta (\alpha \cdot C) + \gamma (\alpha \cdot R_s)$$

$$T_2 = C^W = \alpha \cdot \beta \cdot C$$

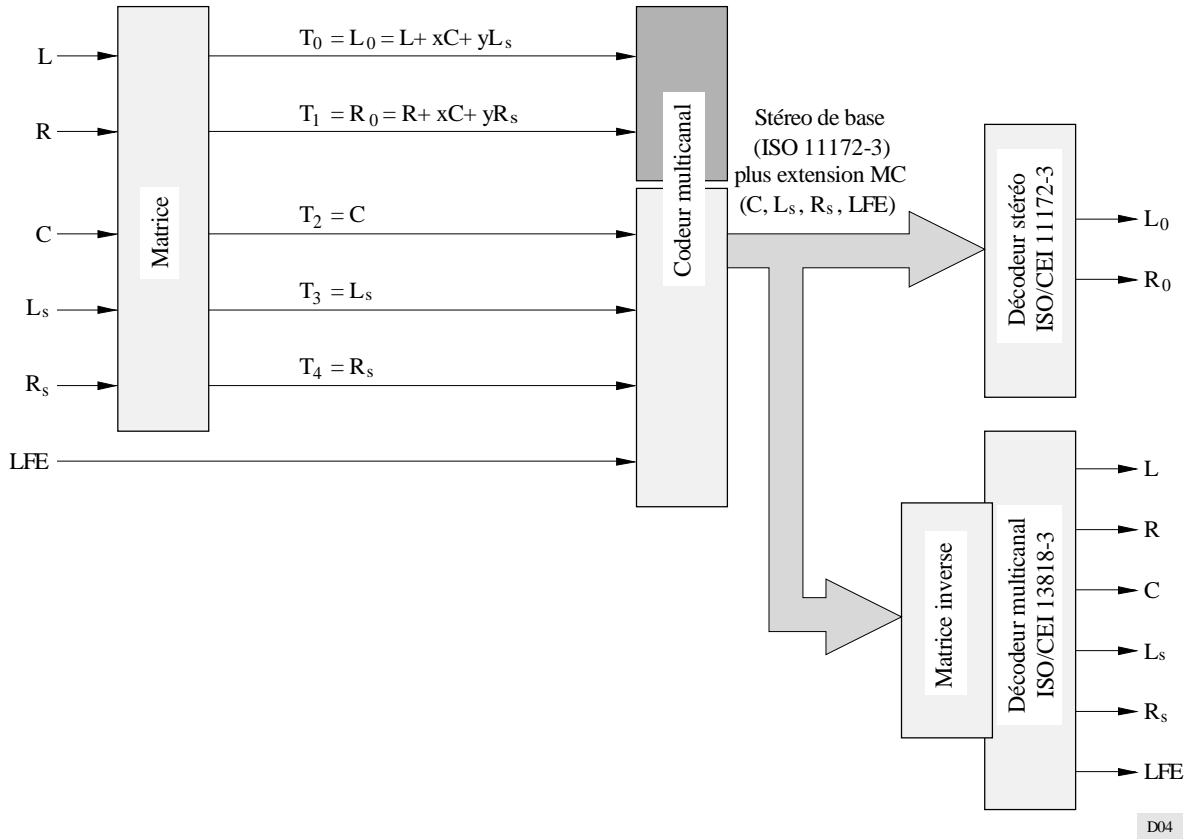
$$T_3 = L_s^W = \alpha \cdot \gamma \cdot L_s$$

$$T_4 = R_s^W = \alpha \cdot \gamma \cdot R_s$$



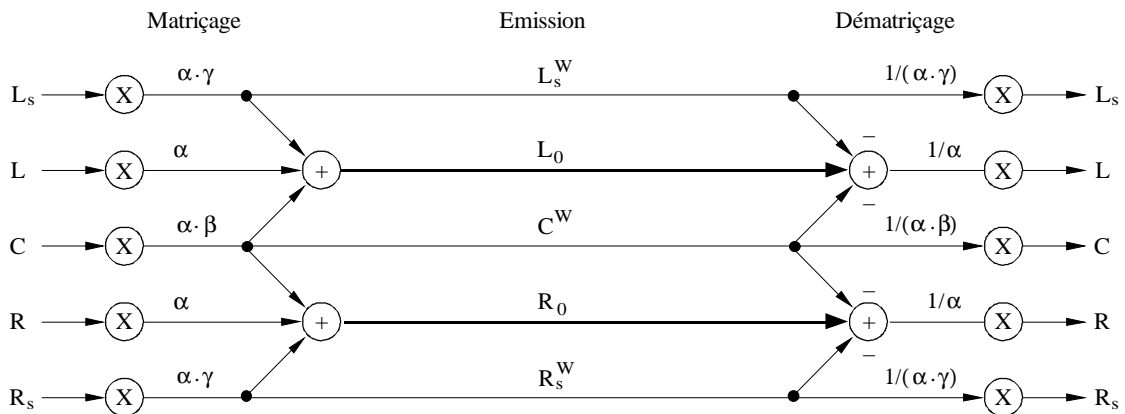
Pour obtenir une réduction maximale du débit binaire, les canaux  $T_2$ ,  $T_3$  et  $T_4$  ont également la possibilité d'acheminer les voies  $(\alpha \cdot L)$  et/ou  $(\alpha \cdot R)$  au lieu des voies  $(\alpha \cdot \beta \cdot C)$ ,  $(\alpha \cdot \gamma \cdot L_s)$  et  $(\alpha \cdot \gamma \cdot R_s)$ .

FIGURE 4  
Rétrocompatibilité du flux audio MPEG-2 avec l'ISO/CEI 11172-3  
en ce qui concerne les informations audio



D04

FIGURE 5  
Matrice d'analyse de compatibilité (dans le codeur) pour créer le signal stéréophonique de base compatible et matrice d'analyse (dans le décodeur) pour établir les cinq canaux audio distincts



D05

Quatre procédures matricielles ont été définies avec des coefficients  $\alpha$ ,  $\beta$  et  $\gamma$ . On peut les choisir dans le codeur multicanal MPEG-2. Trois de ces procédures ajoutent le signal de voie centrale aux signaux gauche (L) et droit (R), avec un affaiblissement de 3 dB. Les signaux de son périphérique,  $L_s$  et  $R_s$ , sont ajoutés aux signaux L ou R avec un affaiblissement de 3 dB ou de 6 dB. La possibilité d'une surcharge des signaux stéréophoniques compatibles  $L_0$  et  $R_0$  est évitée par l'application du facteur d'affaiblissement  $\alpha$  aux signaux individuels L, R, C,  $L_s$  et  $R_s$  avant leur matriciage. L'une de ces procédures assure la compatibilité avec le procédé Dolby Surround ®. Comme il s'agit d'un format bicanal, la compatibilité peut déjà être assurée en trames MPEG-1. La norme MPEG-2 permet de compléter de tels signaux jusqu'à une largeur complète de 5 canaux distincts.

La quatrième procédure implique qu'aucune matrice n'est incluse, ce qui revient en fait à une sorte de mode non rétrocompatible pour le codec multicanal MPEG-2, dans la mesure où un décodeur MPEG-1 produira les signaux L et R de l'assortiment multicanal. Dans certaines conditions d'enregistrement, cette pseudo-matrice contiendra l'assortiment optimal de voies stéréophoniques.

Le terme *précompatibilité* signifie qu'un futur décodeur multicanal doit être capable de décoder correctement le flux binaire stéréophonique de base 2/0.

On réalise cette compatibilité en exploitant le champ de données auxiliaires de la trame audio selon l'ISO/CEI 11172-3 pour fournir des canaux additionnels (voir la Fig. 6). La «longueur variable» du champ de données auxiliaires donne la possibilité d'acheminer l'ensemble des informations d'extension en canaux multiples. Un simple décodeur à deux voies audio MPEG-1 ne pourra qu'ignorer cette partie du champ de données auxiliaires. Si le signal audio multicanal dépasse le débit de 384 kbit/s pour la couche II, une partie d'extension est ajoutée à la partie compatible avec les trames MPEG-1. Cependant, toutes les informations relatives au signal stéréophonique compatible doivent être conservées dans la partie compatible avec les trames MPEG-1. Dans ce cas, la trame audio MPEG-2 se compose de la partie compatible MPEG-1 et de la partie (incompatible) d'extension multicanal, comme indiqué sur la Fig. 7.

Un exemple de cette stratégie est donné par le système RAN EU147 [5] [6] qui n'acheminera pas, dans sa première génération, de signaux multicanaux. L'extension aux canaux audionumériques périphériques doit donc être rétro/précompatible avec les décodeurs audio MPEG-1.

### 3.1.3 Infocompatibilité

Concernant la présentation stéréophonique du signal audio, des groupes d'experts de l'UIT-R, de la SMPTE et de l'UER recommandent un système à 5 canaux en tant que format de référence pour le son périphérique, avec une voie centrale C et deux voies périphériques  $L_s$ ,  $R_s$  en plus des voies stéréophoniques de gauche et de droite, L et R. Ce système est dit *stéréo 3/2* (3 voies frontales et 2 voies périphériques). Il oblige à gérer cinq canaux dans la chaîne du studio, du support d'enregistrement, de la contribution, de la distribution, des liaisons d'émission jusqu'au domicile.

Avec une hiérarchie de formats sonores fournissant un plus petit nombre de voies et une performance réduite en terme de présentation stéréophonique (jusqu'à un minimum de 2 voies stéréo ou même la monophonie) et un système d'équations de réduction de l'assortiment, la couche II audio MPEG-2 assure l'infocompatibilité qui est représentée sur la Fig. 8. Les autres formats sonores de niveau inférieur qui peuvent être utiles sont 3/1, 3/0 ainsi que 2/2, 2/1, 2/0 et 1/0. On peut y faire appel lorsque des contraintes d'ordre économique ou technique (capacité en canaux) s'appliquent à la liaison de transmission ou lorsque l'on ne souhaite qu'un plus petit nombre de voies de reproduction, comme pour la réception de programmes de télévision sur poste portatif.

### 3.1.4 Extension au multilinguisme et aux services associés

Dans les applications de TVHD en particulier, le service principal doit être complété, non seulement de caractéristiques stéréophoniques à plusieurs canaux mais aussi de services associés tels que les programmes bilingues ou les dialogues/commentaires multilingues. La couche II audio MPEG-2 offre diverses configurations de voies sonores dans le système de son multicanal. Par exemple, l'application du deuxième programme stéréophonique peut être un programme stéréophonique bilingue à 2 voies ou la transmission d'un signal binauriculaire supplémentaire. D'autres configurations peuvent consister en un format avec son périphérique 3/2 plus services associés (par exemple, dialogue pur pour les malentendants, commentaire pour les malvoyants, commentaire plurilingue, etc.). Pour ces services, on peut utiliser soit l'extension multilingue ou le champ de données auxiliaires, offerts tous les deux par le flux binaire de couche II MPEG-2.

Un cas évident de service multilingue associé à un son périphérique est celui où la contribution parlée ne fait pas partie de l'environnement acoustique qui est restitué. En d'autres termes, il est relativement facile d'obtenir, pour une émission de sport, des effets sonores périphériques avec voies monophoniques de commentaire en diverses langues. En revanche, un son périphérique dans une émission dramatique nécessiterait un nouvel assortiment de cinq canaux pour chaque langue supplémentaire.

FIGURE 6

Rétrocompatibilité avec l'ISO/CEI 11172-3 et avec la syntaxe des trames audio MPEG- Champ de données auxiliaires de la trame de couche II MPEG-1 acheminant des informations d'extension en multicanaux (MC)

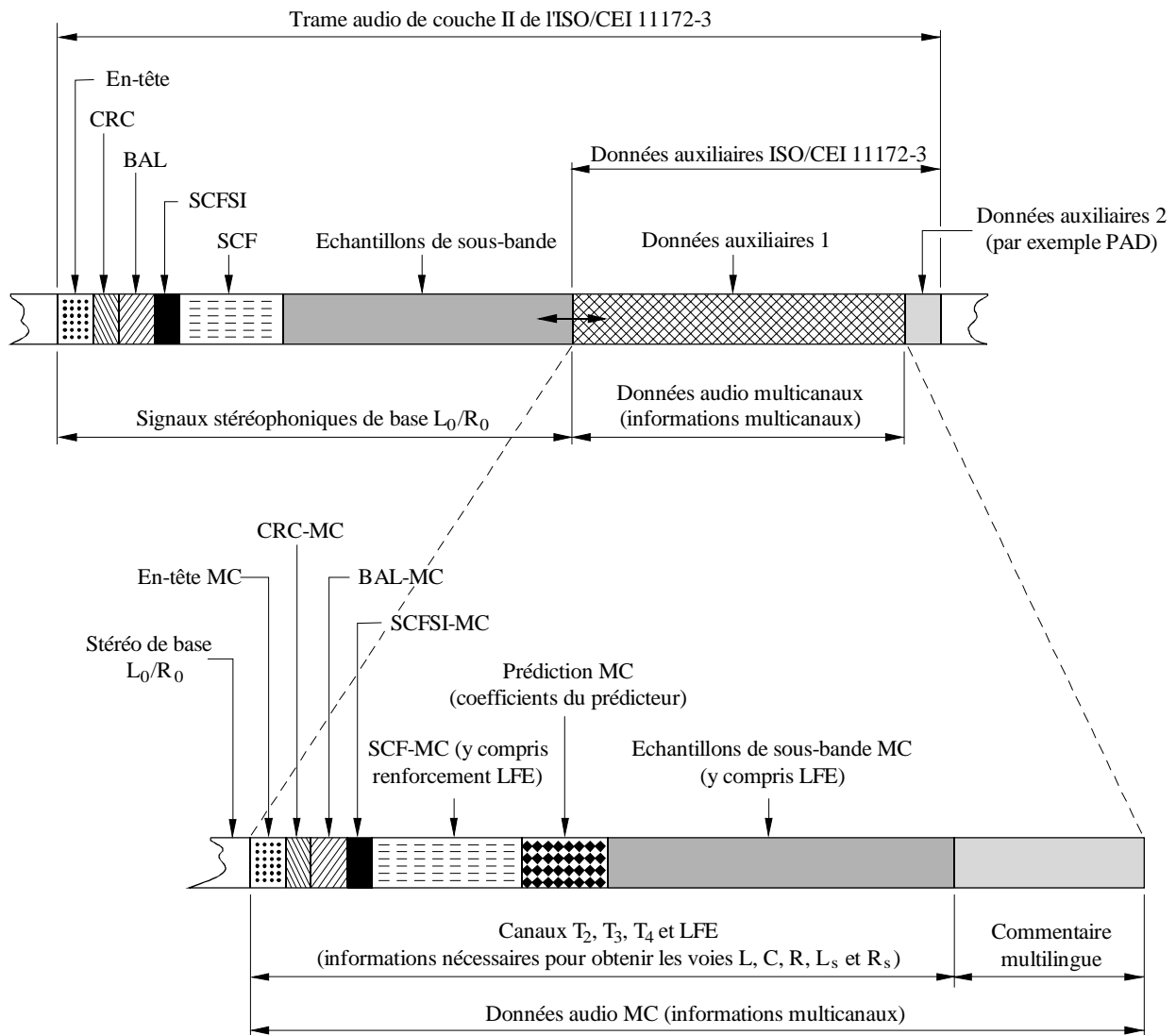
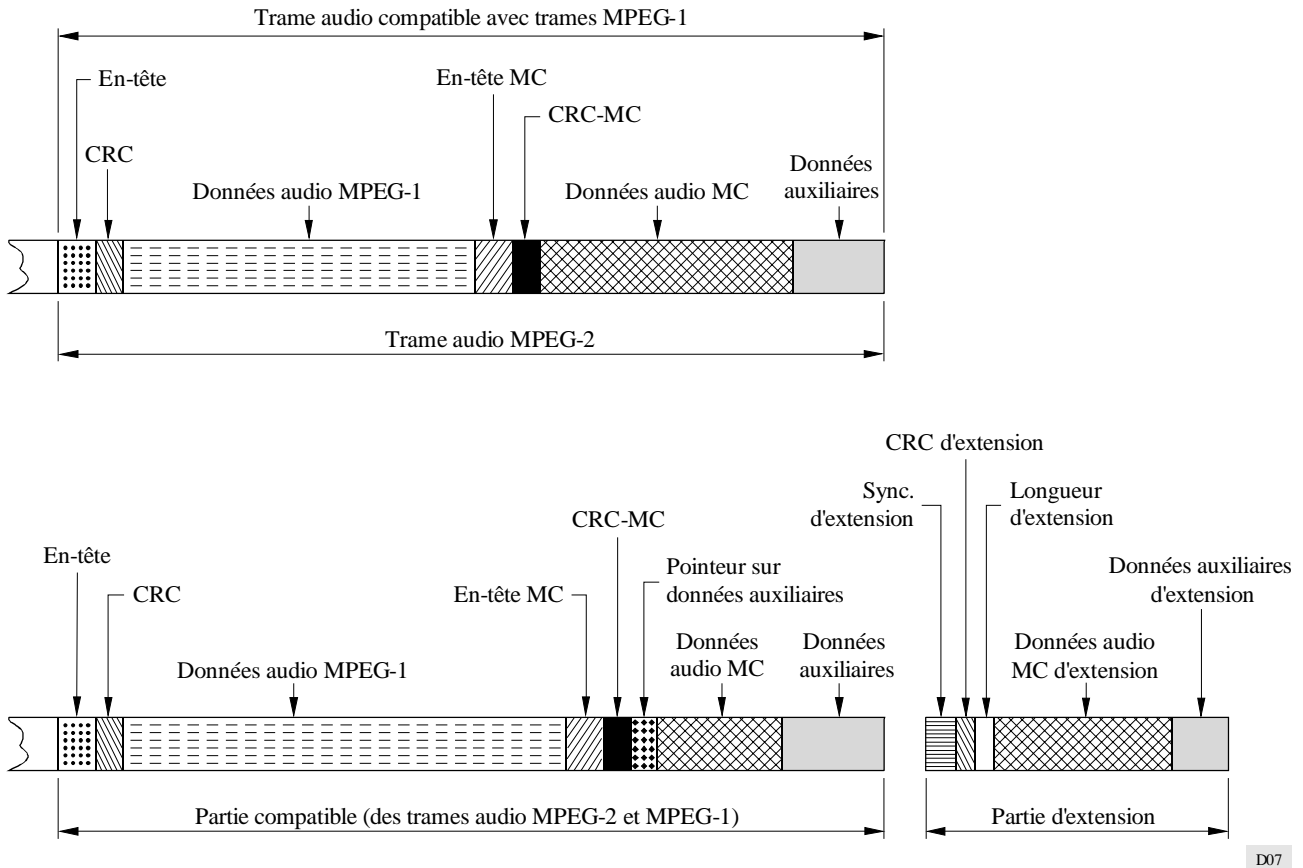


FIGURE 7

Trame audio multicanal MC de couche II de l'ISO/CEI 13818-3 (audio MPEG-2) composée d'une partie compatible avec les trames MPEG-1 et d'une partie d'extension



D07

Un problème important est sans doute celui de l'assortiment final dans le décodeur, c'est-à-dire la reproduction d'un seul commentaire/dialogue choisi (par exemple au moyen du haut-parleur central) associée à l'assortiment commun de musique et d'effets stéréophoniques transmis (comme dans le cas d'un documentaire ou d'un reportage sportif). Si la rétrocompatibilité est requise, les signaux de base doivent contenir les informations correspondant au signal de commentaire/dialogue primaire, qui doivent être soustraites dans le décodeur multicanal en cas de sélection d'un autre commentaire/dialogue.

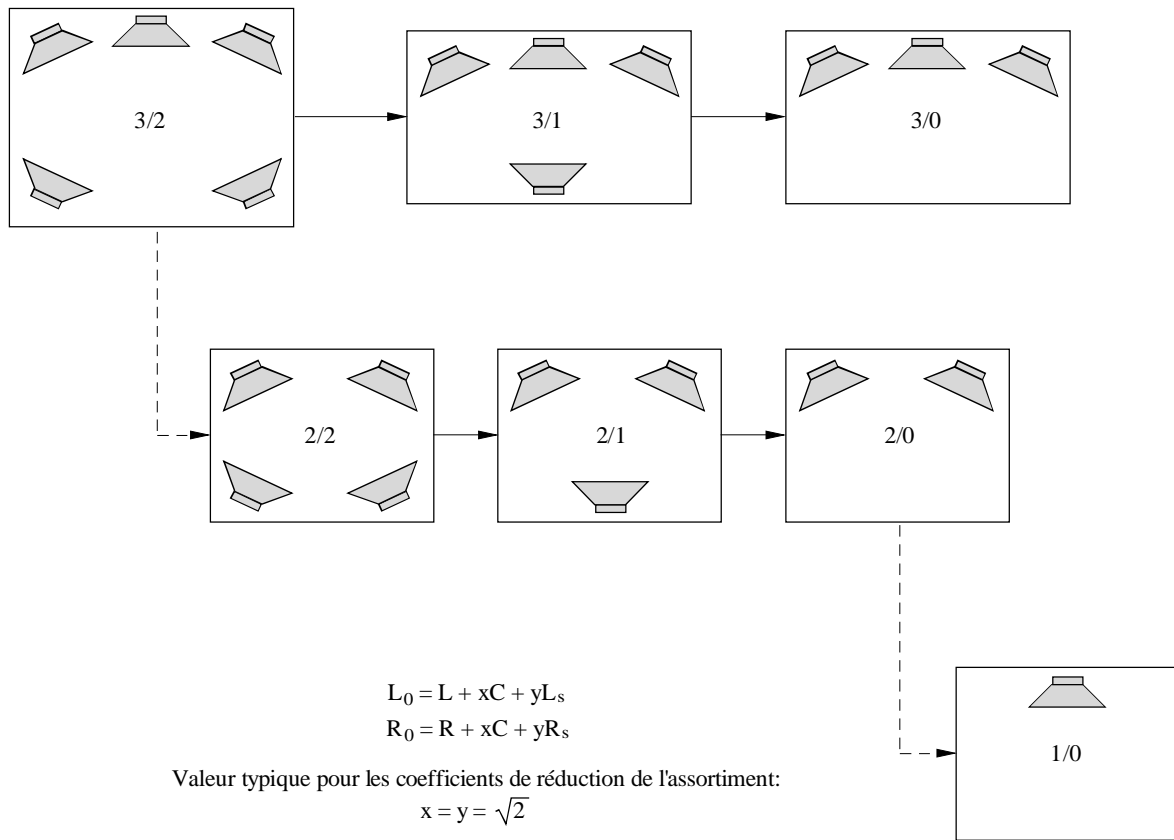
En plus de ces services, il y a lieu que les diffuseurs envisagent les services destinés aux consommateurs malentendants et malvoyants. Dans le cas des malentendants, un canal de dialogue pur (c'est-à-dire sans effets sonores) serait le plus avantageux. Pour les malvoyants, il faudrait un canal descriptif. Dans les deux cas, ces services pourraient être transmis à faible débit binaire (environ 48 kbit/s) avec la technique de codage en fréquence d'échantillonnage inférieure qui assure une excellente qualité des signaux vocaux à un débit binaire de 64 kbit/s et même moins, ce qui exercerait une très faible pression sur la capacité disponible dans le canal de transmission.

### 3.1.5 Canal de renforcement des basses fréquences

Conformément aux projets de nouvelles Recommandations UIT-R du Groupe d'action 10/1 des radiocommunications, le format stéréophonique en 3/2 devrait offrir une option de renforcement des basses fréquences (LFE) pouvant, en plus des canaux principaux à bande complète, acheminer des signaux de fréquence comprise entre 20 et 120 Hz. La fonction de ce canal est de permettre aux auditeurs d'augmenter, s'ils le souhaitent, le contenu en basses fréquences du programme audio, en termes de valeur comme de niveau de ces fréquences. Du point de vue du producteur, cette option permet de régler la marge à un plus bas niveau dans les canaux audio principaux.

FIGURE 8

Options de réduction (de 3/2 à 1/0) des assortiments de voies périphériques audio MPEG-2



D08

## 3.2 Stratégies de codage composite pour les signaux audio multicanaux

Si l'on utilise des méthodes de codage composite pour un programme audio comportant plus d'un seul canal, le débit binaire requis n'augmente pas proportionnellement au nombre de canaux. En mode audio sur plusieurs canaux, la technique de codage composite est très efficace car il existe un grand nombre de corrélations, tant dans le signal proprement dit que dans la perception stéréophonique de ce signal. En codage composite, les portions inapplicables et redondantes des signaux stéréophoniques sont éliminées. Les effets suivants peuvent être utilisés:

### 3.2.1 Diaphonie dynamique

Une certaine portion des signaux stéréophoniques, surtout dans la région des hautes fréquences, ne contribuent pas à la localisation auditive des sources sonores. On peut reproduire cette portion au moyen d'un haut-parleur quelconque. Etant donné que, dans les très hautes fréquences, la localisation des sources dépend plus de la forme spectrale (c'est-à-dire du rapport entre l'énergie du signal et sa fréquence) que des informations de phase, on peut appliquer un codage de puissance stéréophonique. Par rapport au codage stéréophonique mixte ou au codage de puissance stéréophonique définis pour les couches I et II du mode MPEG-1, la diaphonie dynamique représente un procédé beaucoup plus souple pour coder le signal d'extension multicanal du mode MPEG-2. Le spectre des fréquences audio est subdivisé en 12 groupes de sous-bande. Dans chacun de ces groupes, on peut appliquer un cas choisi parmi 15. L'information d'affectation des bits et les échantillons quantifiés d'un, de deux ou des trois canaux de transmission ( $T_2$ ,  $T_3$ ,  $T_4$ ) ne peuvent pas être transmis, mais seuls peuvent l'être les facteurs de normalisation correspondants. Dans le décodeur, les échantillons manquants sont remplacés par ceux du canal de transmission correspondant.

### 3.2.2 Codage fictif du canal central

Le canal central offre une position stable, en particulier pour les signaux audio censés provenir du centre, comme dans un dialogue, surtout dans le cas d'une vaste zone d'écoute. On a montré expérimentalement [7] que l'avantage d'un canal central n'était pas diminué si la bande passante de ce canal était limitée vers le haut à environ 9 kHz, les fréquences plus hautes étant transmises par les voies gauche et droite. Le canal central représente donc un point fictif aux fréquences élevées.

### 3.2.3 Prédiction adaptative en canaux multiples

Certains signaux stéréophoniques contiennent des portions à diaphonie (intercanaux) cohérente qui, en principe, pourraient être transmises dans un seul canal au lieu de deux. Dans le cas de la prédiction en canaux multiples, qui peut être utilisée indépendamment dans chacun des 12 groupes de sous-bande, les signaux  $T_2$ ,  $T_3$  et  $T_4$  sont prédits sur la base des canaux de transmission  $T_0$  et  $T_1$  du signal stéréophonique de base. Au lieu de transmettre les échantillons de sous-bande quantifiés, seule l'erreur de prédiction est transmise, de concert avec les coefficients de prédiction et les informations relatives à la compensation du délai, que l'on peut utiliser pour améliorer l'efficacité. Le gain de prédiction dépend assez étroitement de la structure du signal de sous-bande. Les signaux à hauteur tonale constante offrent un gain beaucoup plus élevé que les transitoires d'un signal audio.

### 3.2.4 Seuil de masquage commun

La capacité de traitement du système auditif est limitée à un certain niveau. Ce système n'est pas en mesure de percevoir certains détails des voies sonores individuelles lors d'une présentation multicanal. L'exploitation du masquage intercanaux peut être faite dans la couche II MPEG-2 sous la forme d'un seuil de masquage commun. Dans le codeur, les seuils individuels (c'est-à-dire de masquage dans le même canal) sont calculés pour chacun des cinq signaux d'entrée audio ( $L$ ,  $C$ ,  $R$ ,  $L_s$ ,  $R_s$ ) de la même façon que dans le codeur stéréophonique de base à algorithme MUSICAM. Les échantillons de sous-bande sont toutefois quantifiés, dans chaque canal, en fonction du plus grand seuil de ce canal et de l'effet de masquage diaphonique (entre canaux), appelé *différence de niveau de masquage*. Cet algorithme se traduit par une diminution du seuil de masquage proportionnelle à la séparation dans l'espace des composantes de masquage.

L'utilisation du seuil de masquage commun à la place du seuil de masquage propre au canal implique que l'on tienne compte de la disposition des haut-parleurs et des dimensions maximales de la zone d'écoute. L'écoute à très faible distance d'un haut-parleur peut aboutir à la perception du bruit de codage. Cet algorithme n'est donc utilisé qu'en cas d'insuffisance notable de capacité binaire. Si les crêtes nécessaires du débit binaire variable dynamiquement sont supérieures au débit disponible, on choisit dans le codeur la combinaison optimale des méthodes de codage par diaphonie dynamique et par seuil de masquage commun.

### 3.2.5 Réserve commune de bits

Le débit binaire requis par canal pour le codage des composantes perceptives dépend du signal. Chaque canal est donc codé en débit binaire variable. La dynamique de cette variation est d'environ 100 kbit/s. Si le débit du flux binaire est appelé à être constant, le débit global de tous les canaux doit être maintenu constant. Etant donné que les débits dynamiques individuels des signaux de la voie centrale et des voies périphériques ne sont pas tout à fait corrélés (pour autant qu'ils le soient), il en résulte un effet de lissage des crêtes du débit global. Cette réserve commune de bits est utilisée par les techniques dites d'échange de bits de la couche II. Elle est particulièrement efficace en mode de codage indépendant.

### 3.2.6 Commutation des canaux de transmission

Alors que les deux signaux stéréophoniques de base,  $L_0$  et  $R_0$ , sont transmis dans les canaux de transmission  $T_0$  et  $T_1$  compatibles avec le mode MPEG-1, toute combinaison des signaux additionnels peut être transmise dans les canaux de transmission  $T_2$ ,  $T_3$  et  $T_4$ . En d'autres termes, la matrice représentée sur la Fig. 2 n'est pas la seule version possible. Le choix d'un sous-ensemble de huit combinaisons possibles est effectué trame par trame pour minimiser le débit binaire global. Ce choix peut être fait dans des groupes de sous-bande individuels, comme dans le cas de la diaphonie dynamique et de la prédiction adaptative multicanal.

## 4 Brève conclusion

Les Normes internationales ISO/CEI 11172-3 et 13818-3 offrent des méthodes efficaces et flexibles de codage audio, ce qui les rend particulièrement appropriées à une large gamme d'applications aux services de télédiffusion. Le sous-groupe Audio MPEG-1 a élaboré une technique de codage pour signaux monophoniques et stéréophoniques dont on peut se servir avec ou sans système de codage des images. Cette technique permet de coder des signaux audio de haute qualité dans la gamme de 192 à environ 100 kbit/s par programme monophonique, assurant une marge suffisante pour mettre en place une cascade et un post-traitement aux débits binaires supérieurs.

L'ISO/CEI 11172-3 a marqué un important jalon dans la première phase d'élaboration d'un codage audio de haute qualité pour usage général dans des applications de diffusion, de télécommunication, d'informatique et d'utilisation à domicile. Mais la mise au point de la norme MPEG-1 n'est pas la fin de la normalisation des systèmes de codage audio à haute fidélité. Le système de codage audio multicanal en trames MPEG-2, qui assure la précompatibilité et la rétrocompatibilité avec les signaux audio codés selon l'ISO/CEI 11172-3, est conçu pour des applications universelles avec ou sans image d'accompagnement. A part la radiodiffusion audionumérique, les applications envisagées sont les systèmes numériques de télévision, les magnétoscopes numériques et les supports d'enregistrement interactifs.

La souplesse de configuration en termes d'affectation des canaux sonores et des débits binaires offre d'utiles combinaisons entre divers niveaux de performance stéréophonique multicanal et divers nombres de canaux, en modes de codage composite et de codage indépendant.

### Références bibliographiques

- [1] ISO/CEI 11172-3 [1992] Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 3: Son. Norme internationale.
- [2] Recommandation UIT-R BS.1115: Codage audio à faible débit binaire.
- [3] Recommandation UIT-T J.52: Transmission numérique de signaux radiophoniques de haute qualité utilisant un, deux ou trois canaux à 64 kbit/s par signal monophonique (et jusqu'à 6 par signal stéréophonique). Doc. CMTT/70, novembre 1993.
- [4] ISO/CEI 13818-3 [novembre 1994] Technologies de l'information – Codage des images animées et du son associé – Partie 3: Son. Norme internationale.
- [5] ETSI [janvier 1995] Norme européenne de télécommunication (NET 300 401): Système de radiodiffusion – Radiodiffusion audionumérique (RAN) vers des récepteurs mobiles, portables et fixes.
- [6] Recommandation UIT-R BS.1114: Systèmes de radiodiffusion sonore numérique de Terre à destination de récepteurs fixes, portatifs ou placés à bord de véhicules, fonctionnant dans la gamme de fréquences 30-3 000 MHz.
- [7] Recommandation UIT-R BS.775: Système de son stéréophonique multicanal avec ou sans image associée.

## ANNEXE 2

### Norme de compression audionumérique (AC-3) (Norme ATSC)

#### TABLE DES MATIÈRES

	Page
Avant-propos.....	18
1 Introduction .....	18
1.1 Objet.....	18
1.2 Codage .....	19
1.3 Décodage.....	20
2 Domaine d'application .....	21

3	Références .....	21
3.1	Normes associées .....	21
3.2	Références informatives .....	22
4	Notation, définitions et terminologie .....	22
4.1	Notation .....	22
4.2	Définitions .....	22
4.3	Abréviations .....	23
5	Syntaxe du flux binaire .....	26
5.1	Trame de synchronisation .....	26
5.2	Sémantique de spécification de la syntaxe .....	27
5.3	Spécification de la syntaxe .....	27
5.3.1	syncinfo – Information de synchronisation .....	27
5.3.2	bsi – Information de flux binaire .....	28
5.3.3	audblk – Bloc audio .....	29
5.3.4	auxdata – Données auxiliaires .....	32
5.3.5	errorcheck – Code de détection d'erreur .....	33
5.4	Description des éléments du flux binaire .....	33
5.4.1	syncinfo – Information de synchronisation .....	33
5.4.2	bsi – Information de flux binaire .....	34
5.4.3	audblk – Bloc audio .....	39
5.4.4	auxdata – Champ de données auxiliaires .....	45
5.4.5	errorcheck – Champ de détection d'erreur de trame .....	47
5.5	Contraintes de flux binaire .....	47
6	Décodage du flux binaire AC-3 .....	47
6.1	Introduction .....	47
6.2	Description générale du processus de décodage .....	48
6.2.1	Flux binaire entrant .....	48
6.2.2	Synchronisation et détection d'erreurs .....	50
6.2.3	Dépaquetage de l'information du flux binaire, information annexe .....	50
6.2.4	Décodage des exposants .....	50
6.2.5	Affectation de bits .....	51
6.2.6	Traitement des mantisses .....	51
6.2.7	Découplage .....	51
6.2.8	Rematriçage .....	51
6.2.9	Compression de plage dynamique .....	51
6.2.10	Transformation inverse .....	51
6.2.11	Fenêtre, superposition/addition .....	51
6.2.12	Sous-mixage .....	51
6.2.13	Tampon de sortie MIC .....	51
6.2.14	Sortie MIC .....	52
7	Description détaillée de l'algorithme .....	52
7.1	Codage des exposants .....	52
7.1.1	Présentation générale .....	52
7.1.2	Stratégie d'exposant .....	52
7.1.3	Décodage des exposants .....	54
7.2	Affectation de bits .....	57
7.2.1	Présentation générale .....	57
7.2.2	Affectation de bits paramétrique .....	58
7.2.3	Tableaux d'affectation de bits .....	63



7.3	Quantification et décodage des mantisses .....	70
7.3.1	Présentation générale .....	70
7.3.2	Développement des mantisses pour une quantification asymétrique ( $6 \leq \text{bap} \leq 15$ ) .....	71
7.3.3	Développement de mantisses pour une quantification symétrique ( $1 \leq \text{bap} \leq 5$ ) .....	71
7.3.4	Bruit de désambiguïsation (dither) pour les mantisses de 0 bit ( $\text{bap} = 0$ ) .....	72
7.3.5	Dégroupage de mantisses .....	74
7.4	Couplage de canal .....	74
7.4.1	Présentation générale .....	74
7.4.2	Structure de sous-bande pour le couplage .....	75
7.4.3	Format de coordonnée de couplage .....	75
7.5	Rematriçage .....	76
7.5.1	Présentation générale .....	76
7.5.2	Définitions des bandes de fréquences .....	77
7.5.3	Technique de codage .....	79
7.5.4	Technique de décodage .....	79
7.6	Normalisation du dialogue .....	79
7.6.1	Présentation générale .....	79
7.7	Compression de plage dynamique .....	80
7.7.1	Commande de plage dynamique; dynrng, dynrng2 .....	80
7.7.2	Compression forte; compr, compr2 .....	82
7.8	Sous-mixage .....	84
7.8.1	Procédure générale de sous-mixage .....	84
7.8.2	Sous-mixage vers deux canaux .....	87
7.9	Equations de transformation et commutation des blocs .....	89
7.9.1	Présentation générale .....	89
7.9.2	Technique .....	89
7.9.3	Réalisation du décodeur .....	89
7.9.4	Equations de la transformation .....	90
7.9.5	Code des plages de gain des canaux .....	94
7.10	Détection d'erreur .....	95
7.10.1	Vérification du CRC .....	95
7.10.2	Vérification de l'homogénéité du flux binaire .....	91
8	Codage du flux binaire AC-3 .....	98
8.1	Introduction .....	98
8.2	Résumé du processus de codage .....	100
8.2.1	Signaux MIC en entrée .....	100
8.2.2	Détection de signaux transitoires .....	100
8.2.3	Transformation directe .....	101
8.2.4	Stratégie de couplage .....	102
8.2.5	Constitution du canal de couplage .....	102
8.2.6	Rematriçage .....	102
8.2.7	Extraction des exposants .....	102
8.2.8	Stratégie d'exposant .....	102
8.2.9	Stratégie de bruit de désambiguïsation (dither) .....	103
8.2.10	Codage des exposants .....	103
8.2.11	Mantisses normalisées .....	103
8.2.12	Affectation centrale de bits .....	103
8.2.13	Quantification des mantisses .....	104
8.2.14	Empaquetage de la trame AC-3 .....	104
	Appendice 1 – Flux élémentaire AC-3 dans un multiplex MPEG-2 .....	104

## Norme de compression audionumérique (AC-3) (Norme ATSC)

### Avant-propos

Dans un souci d'efficacité, le Comité ATSC (Advanced Television System Committee) a été créé par les organisations membres du Comité conjoint de coordination inter-sociétés (JCIC)\*, conscient qu'il était essentiel pour le futur développement de services nationaux de télévision de définir un ensemble de normes nationales coordonnées dans les meilleurs délais.

Une des tâches du Comité ATSC est d'étudier s'il est nécessaire d'élaborer des normes techniques non obligatoires pour les systèmes de télévision évoluée (TVE) et de coordonner le cas d'élaboration de ces normes. Le Comité exécutif du Comité ATSC a confié à un certain nombre de spécialistes travaillant sous la responsabilité du Technology group on distribution (T3), la rédaction de la norme américaine de TVE. L'Audio specialist group (T3/S7) était chargé de rédiger la norme audio de TVE.

La présente Recommandation a été préparée à l'origine par l'Audio specialist group dans le cadre de sa tâche de rédaction de la norme américaine de diffusion pour la TVE. Elle a été approuvée le 26 septembre 1994 par le Technology Group on Distribution et le 10 novembre 1994 comme Norme ATSC par l'ensemble des membres de l'ATSC. L'Appendice 1 de l'Annexe 2, «Flux élémentaires AC-3 dans un multiplex MPEG-2», a été approuvé par le Technology Group on Distribution le 23 février 1995 et par tous les membres de l'ATSC le 12 avril 1995. La Norme ATSC A/53, «Norme de télévision numérique pour la transmission TVHD», fait référence à la présente Recommandation et décrit de quelle manière l'algorithme de codage ci-inclus s'applique dans la norme américaine de TVE.

Au moment de la parution de la présente Recommandation, la description du système qu'il contient n'a pas fait l'objet de vérifications par transmission de signaux provenant de codeurs mis au point de manière indépendante vers des décodeurs mis au point séparément.

## 1 Introduction

### 1.1 Objet

Il est possible de réduire le volume d'informations nécessaire pour représenter des signaux audio, et ceci dans le but de les diffuser ou de les enregistrer d'une manière plus efficace. Dans le cas de signaux audionumériques, le volume d'informations numériques nécessaire pour reproduire avec précision les échantillons d'origine en modulation par impulsions et codage (MIC) peut être réduit en utilisant un algorithme de *compression numérique* du signal (le terme de compression utilisé ici concerne la compression du volume d'informations numériques devant être mémorisées ou enregistrées et non pas la compression de la dynamique du signal audio). Le but de l'algorithme de compression numérique est de donner une représentation numérique du signal audio qui, après décodage, sera perçue de la même façon que le signal d'origine tout en utilisant une information numérique minimale (débit binaire) pour la représentation comprimée (ou codée) du signal. L'algorithme de compression numérique AC-3 spécifié dans la présente Recommandation peut coder de 1 à 5,1 canaux audio, à partir d'une représentation MIC, en un flux binaire série dont le débit est compris entre 32 et 640 kbit/s. Le canal «0,1» désigne un canal de largeur de bande réduite acheminant uniquement les signaux à très basse fréquence.

La Fig. 9 présente un exemple d'utilisation type de cet algorithme. Le programme audio à 5,1 canaux est converti par le codeur AC-3 depuis une représentation MIC à 5 Mbit/s environ (6 canaux  $\times$  48 kHz  $\times$  18 bits = 5,184 Mbit/s) en un flux binaire série à 384 kbit/s. Un équipement de transmission par satellite transforme ce flux binaire en un signal

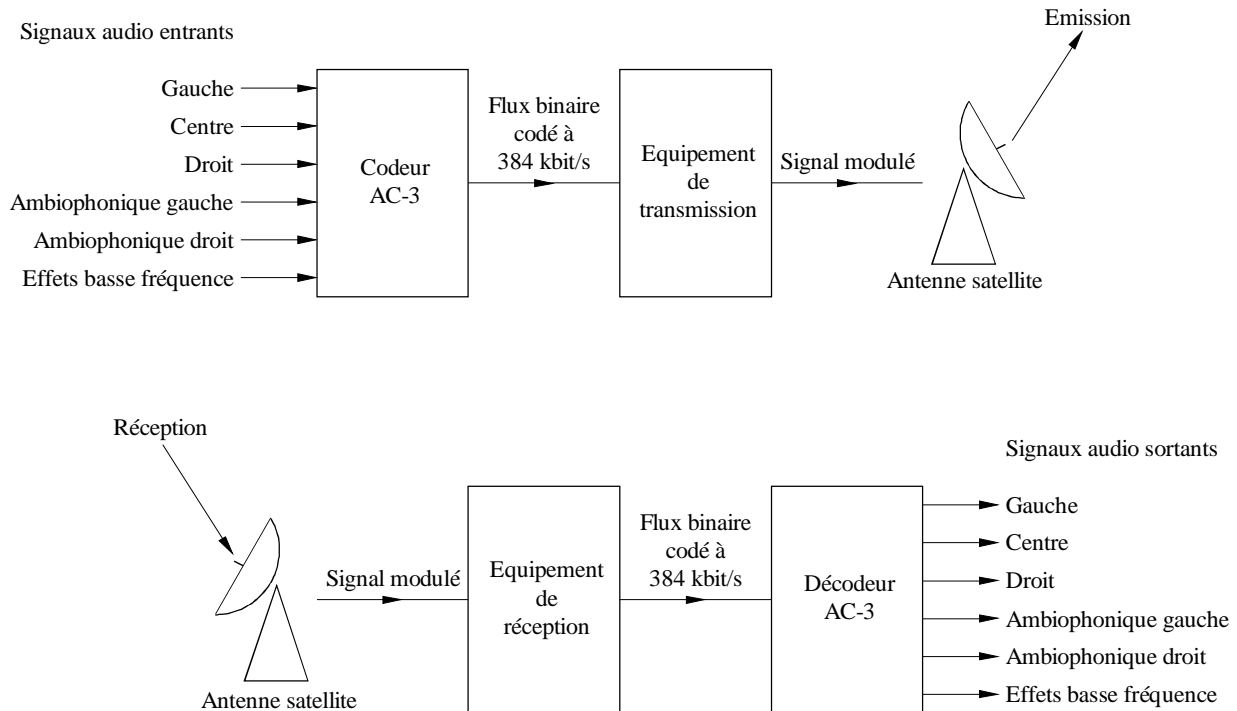
---

\* Le comité JCIC est actuellement composé de l'Electronic Industries Association (EIA), de l'IEEE, de la National Association of Broadcasters (NAB), de la National Cable Television Association (NCTA) et de la Society of Motion Picture and Television Engineers (SMPTE).

NOTE 1 – L'attention du lecteur est attirée sur le fait que la conformité à la présente Norme peut nécessiter l'utilisation d'une invention couverte par des brevets. La publication de la présente Norme n'implique aucun avis à cet égard ni à l'égard de tout autre brevet en rapport avec celui-ci. Toutefois, le détenteur du brevet a déposé une déclaration aux termes de laquelle il est disposé à octroyer une licence d'exploitation, en des termes et conditions raisonnables et non discriminatoires, à tous ceux qui en exprimeront le désir. Pour de plus amples détails, s'adresser à l'éditeur.

radiofréquence qui est émis vers un répéteur de satellite. Le codeur AC-3 permet de réduire la largeur de bande et la puissance nécessaires par un facteur supérieur à 13. Le signal reçu depuis le satellite est démodulé en un flux binaire série à 384 kbit/s puis appliqué au décodeur AC-3. On obtient finalement le programme audio d'origine à 5,1 canaux.

FIGURE 9  
Exemple d'utilisation du codage AC-3 à la transmission audio par satellite



D09

La compression numérique du signal audio est utile chaque fois que la réduction du volume d'informations numériques nécessaire pour la représentation d'un signal audio apporte un avantage économique. La compression peut être utilisée pour la diffusion audio par voie de Terre ou par satellite, par câble ou le stockage de signaux audio sur des supports magnétiques, optiques, à semi-conducteurs ou autres.

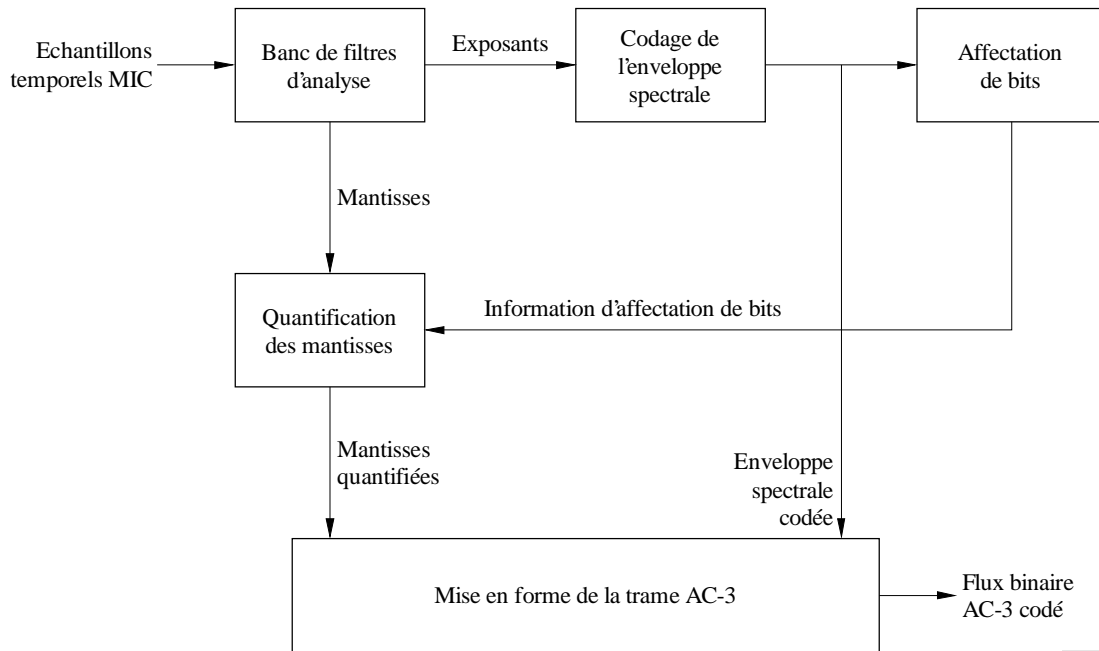
## 1.2 Codage

Le codeur AC-3 accepte des signaux audio MIC et produit un flux binaire codé conforme à la présente Norme. La présente Norme ne spécifie pas le détail du processus de codage. Néanmoins, le codeur doit produire un flux binaire respectant la syntaxe décrite dans le § 5 qui, après un décodage conforme aux § 6 et 7, produit un signal audio de qualité suffisante pour l'application envisagée. Le § 8 contient des informations plus détaillées sur le processus de codage décrit succinctement ci-après.

L'algorithme AC-3 permet un gain de codage élevé (défini comme le rapport taux binaire d'entrée sur taux binaire de sortie) par quantification approximative d'une représentation du signal audio dans le domaine fréquence. La Fig. 10 représente ce processus sous forme synoptique. La première étape du processus de codage consiste à transformer la représentation du signal audio de la séquence d'échantillons temporels MIC en une séquence de blocs de coefficients de fréquence. Ceci est réalisé par le banc de filtres d'analyse. Les blocs de 512 échantillons temporels avec chevauchement partiel sont multipliés par une fenêtre temporelle et subissent une transformation vers le domaine fréquence. En raison du chevauchement, chaque échantillon MIC entrant est représenté dans deux blocs séquentiels transformés. Le domaine fréquence peut alors être réduit dans un rapport 2 de manière que chaque bloc ne contienne que 256 coefficients fréquentiels. Chaque coefficient fréquentiel est représenté en notation exponentielle binaire sous la forme d'un exposant binaire et d'une mantisse. L'ensemble des exposants est codé sous la forme d'une représentation de manière à représenter

approximativement le spectre du signal. Cette représentation appelée enveloppe spectrale, est utilisée par un sous-programme commun d'attribution de bits qui détermine le nombre de bits à utiliser pour coder chaque mantisse. L'enveloppe spectrale et les mantisses approximativement quantifiées de 6 blocs audio (1 536 échantillons audio) sont formatées en une trame AC-3. Le flux binaire AC-3 est une séquence de trames AC-3.

FIGURE 10  
Codeur AC-3



D10

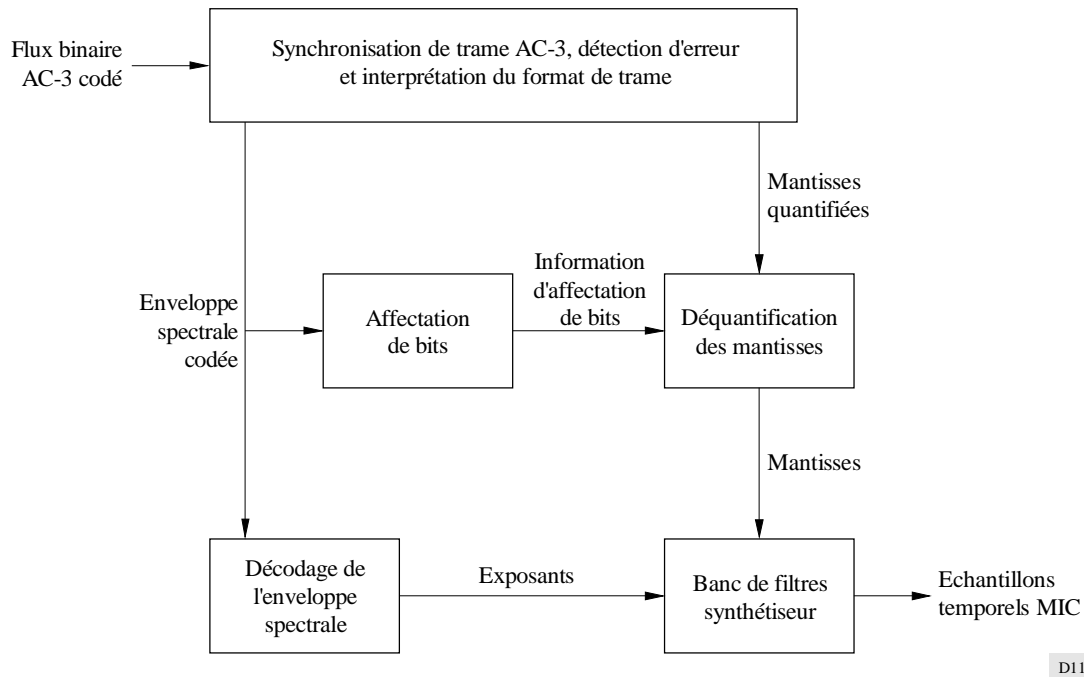
En réalité le fonctionnement du codeur AC-3 est plus complexe que ne l'indique la Fig. 10 car:

- un en-tête de trame est ajouté, contenant les informations nécessaires pour la synchronisation, le codage et le décodage du flux binaire (taux binaire, fréquence d'échantillonnage, nombre de canaux codés, etc.);
- des codes de détection d'erreur sont insérés de manière à permettre au décodeur de vérifier qu'une trame reçue ne contient pas d'erreurs;
- la résolution spectrale du banc de filtres d'analyse peut être modifiée d'une manière dynamique pour assurer une meilleure adéquation des caractéristiques temps/fréquence de chaque bloc audio;
- l'enveloppe spectrale peut être codée avec une résolution temps/fréquence variable;
- une attribution de bits plus complexe peut être réalisée et les paramètres du sous-programme central d'attribution de bits peuvent être modifiés afin d'obtenir une attribution de bits plus optimisée;
- les canaux peuvent être couplés entre eux aux fréquences hautes afin de réaliser un gain de codage plus élevé pour un fonctionnement à des taux binaires plus faibles;
- dans le mode bicanal, un processus de rematriçage peut être exécuté de manière sélective afin d'obtenir un gain de codage supplémentaire dans le cas où le signal bicanal est décodé avec un décodeur ambiophonique matriciel.

### 1.3 Décodage

Le décodage est pratiquement le processus inverse du codage. Le décodeur présenté à la Fig. 11 doit se synchroniser sur le flux binaire codé, détecter les erreurs et interpréter le format des différents types de données telles que l'enveloppe spectrale codée et les mantisses quantifiées. Le sous-programme d'attribution des bits est exécuté et les résultats sont utilisés pour désassembler et déquantifier les mantisses et l'enveloppe spectrale est décodée de manière à produire les exposants. Les exposants et mantisses sont reconvertis dans le domaine temps afin d'obtenir les échantillons temporels MIC décodés.

FIGURE 11  
Décodeur AC-3



D11

En réalité le fonctionnement du décodeur AC-3 est plus complexe que ne l'indique la Fig. 11 car:

- en cas de détection d'erreur le décodeur peut rester muet ou corriger l'erreur;
- les canaux dont les contenus à haute fréquence ont été regroupés doivent être groupés;
- une opération de dématricage doit être exécutée (dans le mode bicanal) lorsque les canaux ont été rematriculés;
- la résolution du banc de filtre synthétiseur doit être dynamiquement modifiée de manière identique à celle avec laquelle le banc de filtres d'analyse du codeur a été modifié pendant le processus de codage.

## 2 Domaine d'application

La présente Norme contient des parties normatives et des parties informatives. Les parties normatives spécifient une représentation codée de l'information audio et spécifient également le processus de décodage. Le codage fait l'objet d'une partie informative. La représentation codée ici spécifiée est adaptée aux applications de transmission et de stockage de signaux numériques audio. La représentation codée peut intégrer de 1 à 5 canaux audio de largeur bande nominale et un canal de renforcement des basses fréquences. La présente spécification est compatible avec une vaste plage de débits binaires codés.

Une dénomination abrégée du présent algorithme de codage est AC-3.

## 3 Références

### 3.1 Normes associées

Les textes référencés ci-dessous contiennent des dispositions auxquelles renvoie la présente Recommandation et qui, de ce fait, sont aussi applicables à la présente Norme. Lors de la parution de la présente Norme, les versions indiquées étaient en vigueur. Toute norme est sujette à révision, et les parties intéressées sont invitées à rechercher la possibilité d'appliquer les versions les plus récentes des normes ci-dessous.

Aucune.

### 3.2 Références informatives

Les textes ci-dessous contiennent des informations sur les algorithmes décrits dans la présente Norme et peuvent être utiles pour les personnes qui utilisent ou étudient la présente Norme. En présence d'informations contradictoires, les informations contenues dans la présente Norme doivent être considérées comme correctes.

TODD, C. et autres [février 1994] AC-3: Flexible Perceptual Coding for Audio Transmission and Storage. 96<sup>e</sup> Convention de l'AES, pré tirage 3796.

EHMER, R.H. [août 1959] Masking Patterns of Tones. *J. Acoust. Soc. Am.*, Vol. 31, 1115-1120.

EHMER, R.H. [septembre 1959] Masking of Tones vs. Noise Bands. *J. Acoust. Soc. Am.*, Vol. 31, 1253-1256.

MOORE, B.C.J. et GLASBERG, B.R. [1987] Formulae Describing Frequency Selectivity as a Function of Frequency and Level, and Their Use in Calculating Excitation Patterns. *Hearing Research*, Vol. 28, 209-225.

ZWICKER, E. [février 1961] Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen). *J. Acoust. Soc. Am.*, Vol. 33, 248.

## 4 Notation, définitions et terminologie

### 4.1 Notation

Dans la présente Recommandation, l'emploi du terme «*doit*» désigne une disposition obligatoire de la présente Norme. L'emploi du terme «*devrait*» désigne une disposition recommandée mais non obligatoire. L'emploi de «*peut*» désigne une caractéristique dont la présence n'exclut pas la conformité et qui peut être présente ou non au choix du réalisateur.

### 4.2 Définitions

Un certain nombre de termes sont utilisés dans la présente Recommandation, dont ceux définis ci-après.

Bande de couplage:	bande de coefficients de transformée de canal de couplage couvrant une ou plusieurs sous-bandes de canal de couplage.
Bin:	le rang du coefficient de fréquence, comme dans «fréquence indice <i>n</i> ». La transformation TDAC à 512 points produit 256 coefficients de fréquence ou bins de fréquence.
Bloc audio:	ensemble de 512 échantillons audio comprenant 256 échantillons du bloc audio précédent et 256 échantillons nouveaux dans le temps. Un nouveau bloc audio apparaît tous les 256 échantillons audio. Un échantillon audio figure donc dans deux blocs audio.
Canal couplé:	canal de largeur de bande complète dont l'information haute fréquence est combinée dans le canal de couplage.
Canal de couplage:	canal formé en combinant l'information haute fréquence des canaux couplés.
Canal d'effets basse fréquence:	canal unique indépendant de largeur de bande limitée (<120 Hz), qui doit être reproduit à un niveau de +10 dB par rapport aux canaux de largeur de bande complète. Le canal d'effets optionnel permet de reproduire des niveaux élevés de pression acoustique pour les sons à basse fréquence.
Canal de largeur de bande complète:	canal audio capable de transmettre la largeur de bande audio complète. Tous les canaux (gauche, centre, droite, ambiophonique gauche et ambiophonique droit) sont des canaux de largeur de bande complète, à l'exception du canal d'effets basse fréquence (lfe)
Canal indépendant:	canal dont l'information haute fréquence n'est pas combinée dans le canal de couplage. (Le canal lfe est toujours indépendant.)
Coefficient:	les échantillons du domaine temps sont convertis par la transformation en coefficients du domaine fréquence.

Ensemble d'exposants:	ensemble constitué par les exposants d'un canal indépendant, du canal de couplage ou de la partie basse fréquence d'un canal couplé.
Enveloppe spectrale:	estimation spectrale constituée de l'ensemble des exposants obtenus par décodage des exposants codés. Similaire (mais non identique) à l'ensemble d'exposants d'origine.
Fenêtre:	vecteur temps que multiplie un bloc audio pour produire un bloc audio avec fenêtrage. La taille de la fenêtre détermine la sélectivité en fréquence du banc de filtres ainsi que les caractéristiques de superposition/addition permettant d'éviter l'apparition de défauts de bloc.
Sous-bande couplage:	sous-bande constituée d'un groupe de coefficients de transformée de 12 canaux de couplage.
Sous-mixage:	combinaison (ou sous-mixage) du contenu de $n$ canaux d'origine produisant $m$ canaux avec $m < n$ .
Trame de synchronisation:	unité du flux binaire séquentiel pouvant être complètement décodée. La trame de synchronisation débute par un code de synchronisation et contient 1 536 échantillons audio codés.

### 4.3 Abréviations

Un certain nombre d'abréviations utilisées dans la présente Recommandation renvoient à des éléments du format AC-3. Le tableau ci-dessous donne les correspondances entre chaque abréviation et les termes ou l'expression qu'elle désigne ainsi que dans la plupart des cas les paragraphes qui les définissent. Ces abréviations sont amplement utilisées dans la présente Recommandation. Les abréviations sont écrites en minuscule et comportent 12 caractères au maximum, elles peuvent ainsi être utilisées dans un codage de logiciel en langage de haut niveau ou en assembleur. Les personnes souhaitant appliquer la présente Norme sont invitées à utiliser les mêmes abréviations dans les codes sources informatiques ou la documentation accompagnant toute réalisation matérielle ou logicielle.

Abréviation	Terminologie	Référence
acmod	Mode de codage audio	§ 5.4.2.3
addbsi	Information supplémentaire de flux binaire	§ 5.4.2.31
addbsie	Présence de l'information supplémentaire de flux binaire	§ 5.4.2.29
addbsil	Longueur de l'information supplémentaire de flux binaire	§ 5.4.2.30
audblk	Bloc audio	§ 5.4.3
audprodie	Présence de l'information de production audio	§ 5.4.2.13
audprodi2e	Présence de l'information de production audio, canal2	§ 5.4.2.21
auxbits	Bits de données auxiliaires	§ 5.4.4.1
auxdata	Champ de données auxiliaires	§ 5.4.4.1
auxdatae	Présence de données auxiliaires	§ 5.4.4.3
auxdata1	Longueur des données auxiliaires	§ 5.4.4.2
baie	Présence de l'information d'allocation de bits	§ 5.4.3.30
bap	Pointeur d'allocation de bits	
bin	Coefficient de fréquence bin dans l'index [bin]	§ 5.4.3.13
blk	Bloc dans l'index de tableau [blk]	
blksw	Indicateur de commutation de bloc	§ 5.4.3.1
bnd	Bande dans l'index de tableau [bnd]	
bsi	Information de flux binaire	§ 5.4.2
bsid	Identification du flux binaire	§ 5.4.2.1
bsmod	Mode de flux binaire	§ 5.4.2.2
ch	Canal dans l'index de tableau [ch]	

Abréviations	Terminologie	Référence
chbwcod	Code de largeur de bande canal	§ 5.4.3.24
chexpstr	Stratégie d'exposant de canal	§ 5.4.3.22
chincpl	Canal couplé	§ 5.4.3.9
chmant	Mantisses de canal	§ 5.4.3.61
clev	Coefficient de niveau de mixage central	§ 5.4.2.4
cmixlev	Niveau du mixage central	§ 5.4.2.4
compr	Mot de gain de compression	§ 5.4.2.10
compr2	Mot de gain de compression, canal2	§ 5.4.2.18
compre	Présence du mot de gain de compression	§ 5.4.2.9
compr2e	Présence du mot de gain de compression, canal2	§ 5.4.2.17
copyrightb	Bit de droit d'auteur	§ 5.4.2.24
cplabsexp	Exposant absolu de couplage	§ 5.4.3.25
cplbegf	Code de fréquence de début de couplage	§ 5.4.3.11
cplbndstrc	Structure de la bande couplage	§ 5.4.3.13
cplco	Coordonnée de couplage	§ 7.4.3
cplcoe	Présence des coordonnées de couplage	§ 5.4.3.14
cplcoexp	Exposant de la coordonnée de couplage	§ 5.4.3.16
cplcomant	Mantisse de la coordonnée de couplage	§ 5.4.3.17
cpldeltba	dba de couplage	§ 5.4.3.53
cpldeltbae	Présence de dba de couplage	§ 5.4.3.48
cpldeltlen	Longueur de dba de couplage	§ 5.4.3.52
cpldeltseg	Nombre de segments de dba de couplage	§ 5.4.3.50
cpldeltfst	Adresse de dba de couplage	§ 5.4.3.51
cplendf	Code de fréquence de fin de couplage	§ 5.4.3.12
cplexps	Exposants de couplage	§ 5.4.3.26
cplexpstr	Stratégie d'exposant de couplage	§ 5.4.3.21
cplfgaincod	Code de couplage de gain rapide	§ 5.4.3.39
cplfleak	Initialisation de fuite rapide couplage	§ 5.4.3.45
cplfsnrofst	Déplacement fin du couplage du rapport S/N	§ 5.4.3.38
cplinu	Couplage utilisé	§ 5.4.3.8
cplleake	Présence de l'initialisation de fuite de couplage	§ 5.4.3.44
cplmant	Mantisses de couplage	§ 5.4.3.61
cplsleak	Initialisation de fuite lente de couplage	§ 5.4.3.46
cplstre	Présence de la stratégie de couplage	§ 5.4.3.7
crc1	CRC – contrôle de redondance cyclique mot 1	§ 5.4.1.2
crc2	CRC – contrôle de redondance cyclique mot 2	§ 5.4.5.2
crcrsv	Bit réservé au CRC	§ 5.4.5.1
csnrofst	Décalage du rapport S/N approximatif	§ 5.4.3.37
d15	Mode codage d'exposant d15	§ 5.4.3.21
d25	Mode codage d'exposant d25	§ 5.4.3.21
d45	Mode codage d'exposant d45	§ 5.4.3.21
dba	Allocation de bits delta	§ 5.4.3.47
dbpbcod	Code de dB par bit	§ 5.4.3.34
deltba	dba de canal	§ 5.4.3.57
deltbae	Présence de dba de canal	§ 5.4.3.49
deltbaie	Présence de l'information dba	§ 5.4.3.47
deltlen	Longueur de dba de canal	§ 5.4.3.56
deltseg	Nombre de segments de dba de canal	§ 5.4.3.54



Abréviations	Terminologie	Référence
deltfst	Adresse de dba de canal	§ 5.4.3.55
dialnorm	Mot de normalisation de dialogue	§ 5.4.2.8
dialnorm2	Mot de normalisation de dialogue, canal2	§ 5.4.2.16
dithflag	Indicateur de bruit de désambiguïsation (dither)	§ 5.4.3.2
dsurmod	Mode ambiophonique Dolby	§ 5.4.2.6
dynrng	Mot de plage de gain dynamique	§ 5.4.3.4
dynrng2	Mot de plage de gain dynamique, canal2	§ 5.4.3.6
dynrngc	Présence du mot de plage de gain dynamique	§ 5.4.3.3
dynrng2c	Présence du mot de plage de gain dynamique, canal2	§ 5.4.3.5
exps	Exposants de canal	§ 5.4.3.27
fbw	Largeur de bande totale	
fdccod	Code de décroissance rapide	§ 5.4.3.32
fgaincod	Code de gain rapide du canal	§ 5.4.3.41
floorcod	Code de seuil de masquage	§ 5.4.3.35
floortab	Table de seuil de masquage	§ 7.2.2.7
frmsizecod	Code de longueur de trame	§ 5.4.1.4
fscod	Code de fréquence d'échantillonnage	§ 5.4.1.3
fsnrffst	Déplacement fin du rapport S/N du canal	§ 5.4.3.40
gainrng	Code de plage de gain de canal	§ 5.4.3.28
grp	Groupe dans l'index de table [grp]	
langcod	Code de langue	§ 5.4.2.12
langcod2	Code de langue, canal2	§ 5.4.2.20
langcode	Présence du code de langue	§ 5.4.2.11
langcod2c	Présence du code de langue, canal2	§ 5.4.2.19
lfe	Effets basse fréquence	
lfeexps	Exposants lfe	§ 5.4.3.29
lfeexpstr	Stratégie lfe	§ 5.4.3.23
lfefgaincod	Code de gain rapide lfe	§ 5.4.3.43
lfefsnrffst	Déplacement fin du rapport S/N d'effets lfe	§ 5.4.3.42
lfemant	Mantisse lfe	§ 5.4.3.63
lfeon	Effet lfe actif	§ 5.4.2.7
mixlevel	Niveau du mixage	§ 5.4.2.14
mixlevel2	Niveau du mixage, canal2	§ 5.4.2.22
mstrcplco	Coordonnée directrice de couplage	§ 5.4.3.15
nauxbits	Nombre de bits auxiliaires	§ 5.4.4.1
nchans	Nombre de canaux	§ 5.4.2.3
nchgrps	Nombre de groupes d'exposants de canal fbw	§ 5.4.3.27
nchmant	Nombre de mantisses de canal fbw	§ 5.4.3.61
ncplbnd	Nombre de bandes couplées structurées	§ 5.4.3.13
ncplgrps	Nombre de groupes d'exposants structurés	§ 5.4.3.26
ncplmant	Nombre de mantisses couplées	§ 5.4.3.62
ncplsubnd	Nombre de sous-bandes de couplage	§ 5.4.3.12
nfchans	Nombre de canaux fbw	§ 5.4.2.3
nlfegrps	Nombre d'exposants de canal lfe	§ 5.4.3.29
nlfemant	Nombre de mantisses de canal lfe	§ 5.4.3.63
origbs	Flux binaire d'origine	§ 5.4.2.25
phsflg	Indicateur de phase	§ 5.4.3.18
phsflginu	Indicateurs de phase utilisés	§ 5.4.3.10

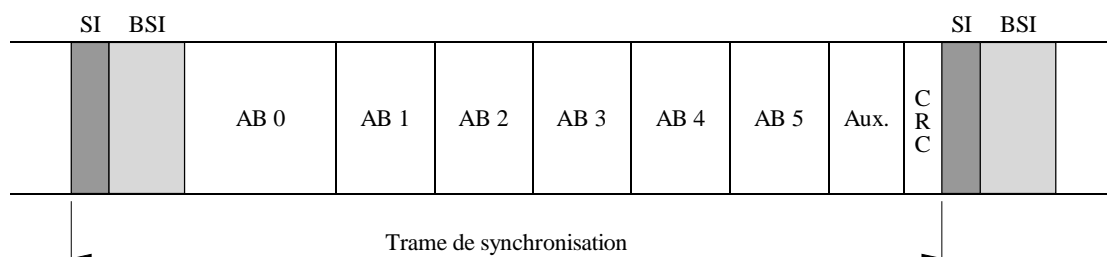
Abréviation	Terminologie	Référence
rbnd	Bande de rematriçage dans l'index de tableau [rbnd]	
rematflg	Indicateur de rematriçage	§ 5.4.3.20
rematstr	Stratégie de rematriçage	§ 5.4.3.19
roomtyp	Type de salle	§ 5.4.2.15
roomtyp2	Type de salle, canal2	§ 5.4.2.23
sbnd	Sous-bande dans l'index de table [sbnd]	
sdycod	Code d'affaiblissement lent	§ 5.4.3.31
seg	Segment dans l'index [seg]	
sgaincod	Code de gain lent	§ 5.4.3.33
skipfld	Champ sauté	§ 5.4.3.60
skipl	Longueur de saut	§ 5.4.3.59
skiple	Présence de la longueur de saut	§ 5.4.3.58
slev	Coefficient de niveau de mixage ambiophonique	§ 5.4.2.5
snroffste	Présence du déplacement du rapport S/N	§ 5.4.3.36
surmixlev	Niveau du mixage ambiophonique	§ 5.4.2.5
syncframe	Trame de synchronisation	§ 5.1
syncinfo	Information de synchronisation	§ 5.3.1
syncword	Mot de synchronisation	§ 5.4.1.1
tdac	Suppression de l'aliasage dû à la division temporelle	
timecod1	Première moitié du code temporel	§ 5.4.2.27
timecod2	Deuxième moitié du code temporel	§ 5.4.2.28
timecod1e	Présence première moitié du code temporel	§ 5.4.2.26
timecod2e	Présence deuxième moitié du code temporel	§ 5.4.2.26

## 5 Syntaxe du flux binaire

### 5.1 Trame de synchronisation

Un flux binaire série audio codé AC-3 est constitué d'une séquence de trames de synchronisation (voir la Fig. 12). Chaque trame de synchronisation contient 6 blocs audio codés (AB), représentant chacun 256 nouveaux échantillons audio. Au début de chaque trame se trouve un en-tête d'information de synchronisation (SI) contenant l'information nécessaire à la synchronisation et à son maintien. L'en-tête SI est suivi d'un en-tête d'information de flux binaire (BSI) contenant des paramètres qui décrivent le service audio codé. Les blocs audio codés peuvent être suivis par un champ de données auxiliaires (Aux.). A la fin de chaque trame se trouve un champ de détection d'erreur qui inclut un mot CRC pour la détection d'erreur. Un mot CRC additionnel, d'utilisation optionnelle, se trouve dans l'en-tête SI.

FIGURE 12  
Trame de synchronisation AC-3



## 5.2 Sémantique de spécification de la syntaxe

Le pseudocode ci-après décrit l'ordre d'arrivée des informations dans le flux binaire. Le pseudocode est basé sur une syntaxe proche de celle du langage C, mais simplifiée pour en faciliter la lecture. Pour les éléments du flux binaire de taille supérieure à 1 bit, l'ordre des bits dans le flux binaire série est, soit le bit de plus fort poids en tête (pour les valeurs numériques), soit le bit de gauche en tête (pour les valeurs de champs binaires). Les champs ou éléments constitutifs du flux binaire sont indiqués en caractères **gras**. Les éléments syntaxiques sont typographiquement signalés par l'utilisation d'un jeu différent (par exemple *dynrng*).

Quelques éléments d'un flux binaire AC-3 se présentent spontanément sous forme de tableaux. La présente spécification de syntaxe traite d'une manière individuelle tous les éléments du flux binaire, qu'ils appartiennent ou non à des tableaux. Les tableaux sont en conséquence décrits comme des éléments multiples (comme dans *blksw[ch]* en opposition avec simplement *blksw* ou *blksw[]*); des structures de commande telles que des boucles *for* sont utilisées pour incrémenter l'index (*[ch]* = canal dans cet exemple).

## 5.3 Spécification de la syntaxe

Un flux binaire audio continu est représenté par une séquence de trames de synchronisation:

Syntaxe
<pre> AC-3_bitstream() {     while(true)     {         <b>syncframe()</b> ;     } } /* fin du flux binaire AC-3 */                 </pre>

La trame de synchronisation, *syncframe*, est constituée des champs **syncinfo** et **bsi**, des 6 champs codés **audblk**, du champ **auxdata** et du champ **errorcheck**.

Syntaxe
<pre> syncframe() {     <b>syncinfo()</b> ;     <b>bsi()</b> ;     for(blk = 0; blk &lt; 6; blk++)     {         <b>audblk()</b> ;     }     <b>auxdata()</b> ;     <b>errorcheck()</b> ; } /* fin de syncframe */                 </pre>

Chaque élément du flux binaire et sa longueur sont détaillés dans le pseudocode qui suit. Il est à noter que tous les éléments du flux binaire arrivent dans le temps avec le bit de plus fort poids en tête ou le bit le plus à gauche en tête.

### 5.3.1 syncinfo – Information de synchronisation

Syntaxe	Taille de mot
<pre> syncinfo() {     <b>syncword</b> .....16     <b>crc1</b> .....16     <b>fscod</b> .....2     <b>frmsizecod</b> .....6 } /* fin de syncinfo */                 </pre>	

## 5.3.2 bsi – Information de flux binaire

Syntaxe	Taille de mot
bsi()	
{	
<b>bsid</b> .....	5
<b>bsmod</b> .....	3
<b>acmod</b> .....	3
if((acmod & 0x1) && (acmod != 0x1)) /* si trois canaux avant */ <b>{cmixlev}</b> .....	2
if(acmod & 0x4) /* si un canal ambiophonique existe */ <b>{surmixlev}</b> .....	2
if(acmod == 0x2) /* si mode 2/0 actif */ <b>{dsurmod}</b> .....	2
<b>lfeon</b> .....	1
<b>dialnorm</b> .....	5
<b>compre</b> .....	1
if(compre) <b>{compr}</b> .....	8
<b>langcode</b> .....	1
if(langcode) <b>{langcod}</b> .....	8
<b>audprodie</b> .....	1
if(audprodie)	
{	
<b>mixlevel</b> .....	5
<b>roomtyp</b> .....	2
}	
if(acmod == 0) /* si mode 1+1 (double mono, certains items nécessitent une seconde valeur) */	
{	
<b>dialnorm2</b> .....	5
<b>compr2e</b> .....	1
if(compr2e) <b>{compr2}</b> .....	8
<b>lngcod2e</b> .....	1
if(lngcod2e) <b>{langcod2}</b> .....	8
<b>audprodi2e</b> .....	1
if(audprodi2e)	
{	
<b>mixlevel2</b> .....	5
<b>roomtyp2</b> .....	2
}	
}	
<b>copyrightb</b> .....	1
<b>origbs</b> .....	1
<b>timecod1e</b> .....	1
if(timecod1e) <b>{timecod1}</b> .....	14
<b>timecod2e</b> .....	1
if(timecod2e) <b>{timecod2}</b> .....	14
<b>addbsie</b> .....	1
if(addbsie)	
{	
<b>addbsil</b> .....	6
<b>addbsi</b> .....	(addbsil+1)×8
}	
}	
} /* fin de bsi */	

## 5.3.3 audblk – Bloc audio

Syntaxe	Taille de mot
audblk() { /* Champs de commutation de blocs et d'indicateurs de bruit de désambiguïisation (dither) */ for(ch = 0; ch < nfchans; ch++) <b>{blksw[ch]}</b> ..... 1 for(ch = 0; ch < nfchans; ch++) <b>{dithflag[ch]}</b> ..... 1  /* Champs de commande dynamique de plage */ <b>dynrng</b> ..... 1 if(dynrng) <b>{dynrng}</b> ..... 8 if(acmod == 0) /* si mode 1+1 */ { <b>dynrng2e</b> ..... 1 if(dynrng2e) <b>{dynrng2}</b> ..... 8 }  /* Champs d'information de stratégie de couplage */ <b>cplstre</b> ..... 1 if(cplstre) { <b>cplinu</b> ..... 1 if(cplinu) { for(ch = 0; ch < nfchans; ch++) <b>{chincpl[ch]}</b> ..... 1 if(acmod == 0x2) <b>{phsflginu}</b> /* si mode 2/0 actif */ ..... 1 <b>cplbegf</b> ..... 4 <b>cplendf</b> ..... 4 /* ncplsubnd = 3 + cplendf - cplbegf */ for(bnd = 1; bnd < ncplsubnd; bnd++) <b>{cplbndstrc[bnd]}</b> ..... 1 } }  /* Champs de coordonnées de couplage et d'indicateurs de phase */ if(cplinu) { for(ch = 0; ch < nfchans; ch++) { if(chincpl[ch]) { <b>cplcoe[ch]</b> ..... 1 if(cplcoe[ch]) { <b>mstrcplco[ch]</b> ..... 2 /* ncplbnd calculé à partir de ncplsubnd et cplbndstrc */ for(bnd = 0; bnd < ncplbnd; bnd++) { <b>cplcoexp[ch][bnd]</b> ..... 4 <b>cplcomant[ch][bnd]</b> ..... 4 } } } } } if((acmod == 0x2) && phsflginu && (cplcoe[0]    cplcoe[1])) { for(bnd = 0; bnd < ncplbnd; bnd++) <b>{phsflg[bnd]}</b> ..... 1 } }	

Syntaxe	Taille de mot
/* Champs pour rematriçage dans le mode 2/0 */	
if(acmod == 0x2) /* si mode 2/0 actif */	
{	
<b>rematstr</b> .....	1
if(rematstr)	
{	
if((cplbegf > 2)    (cplinu == 0))	
{	
for(rbnd = 0; rbnd < 4; rbnd++) <b>{rematflg[rbnd]}</b> .....	1
}	
if((2 ≥ cplbegf > 0) && cplinu)	
{	
for(rbnd = 0; rbnd < 3; rbnd++) <b>{rematflg[rbnd]}</b> .....	1
}	
if((cplbegf == 0) && cplinu)	
{	
for(rbnd = 0; rbnd < 2; rbnd++) <b>{rematflg[rbnd]}</b> .....	1
}	
}	
}	
}	
/* Champs pour la stratégie d'exposants */	
if(cplinu) <b>{cplexpstr}</b> .....	2
for(ch = 0; ch < nfchans; ch++) <b>{chexpstr[ch]}</b> .....	2
if(lfeon) <b>{lfeexpstr}</b> .....	1
for(ch = 0; ch < nfchans; ch++)	
{	
if(chexpstr[ch] != reuse)	
{	
if(!chincpl[ch]) <b>{chbwcod[ch]}</b> .....	6
}	
}	
/* Champs pour les exposants */	
if(cplinu) /* exposants du canal de couplage */	
{	
if(cplexpstr != reuse)	
{	
<b>cplabsexp</b> .....	4
/* ncplgrps calculé à partir de ncplsubnd et de cplexpstr */	
for(grp = 0; grp < ncplgrps; grp++) <b>{cplexps[grp]}</b> .....	7
}	
}	
for(ch = 0; ch < nfchans; ch++) /* exposants pour les canaux de largeur de bande totale */	
{	
if(chexpstr[ch] != reuse)	
{	
<b>exps[ch][0]</b> .....	4
/* nchgrps calculé à partir de chexpstr[ch], et cplbegf ou chbwcod[ch] */	
for(grp = 1; grp <= nchgrps[ch]; grp++) <b>{exps[ch][grp]}</b> .....	7
<b>gainrng[ch]</b> .....	2
}	
}	
if(lfeon) /* exposants du canal d'effets basse fréquence */	
{	
if(lfeexpstr != reuse)	
{	
<b>lfeexps[0]</b> .....	4
/* nlfegrps = 2 */	
for(grp = 1; grp <= nlfegrps; grp++) <b>{lfeexps[grp]}</b> .....	7
}	
}	

Syntaxe	Taille de mot
/* Champs pour l'information paramétrique d'allocation de bits */	
<b>baie</b> .....	1
if(baie)	
{	
<b>sdccod</b> .....	2
<b>fdccod</b> .....	2
<b>sgaincod</b> .....	2
<b>dbpbcod</b> .....	2
<b>floorcod</b> .....	3
}	
<b>snroffste</b> .....	1
if(snroffste)	
{	
<b>csnroffst</b> .....	6
if(cplinu)	
{	
<b>cplfsnroffst</b> .....	4
<b>cplfgaincod</b> .....	3
}	
for(ch = 0; ch < nfchans; ch++)	
{	
<b>fsnroffst[ch]</b> .....	4
<b>fgaincod[ch]</b> .....	3
}	
if(lfeon)	
{	
<b>lfe snroffst</b> .....	4
<b>lfe gaincod</b> .....	3
}	
}	
if(cplinu)	
{	
<b>cplleake</b> .....	1
if(cplleake)	
{	
<b>cplfleak</b> .....	3
<b>cplsleak</b> .....	3
}	
}	
/* Champs pour l'information d'allocation de bits delta */	
<b>deltbaie</b> .....	1
if(deltbaie)	
{	
if(cplinu) <b>{cpldeltbae}</b> .....	2
for(ch = 0; ch < nfchans; ch++) <b>{deltbae[ch]}</b> .....	2
if(cplinu)	
{	
if(cpldeltbae==new info follows)	
{	
<b>cpldeltseg</b> .....	3
for(seg = 0; seg <= cpldeltseg; seg++)	
{	
<b>cpldeltoffst[seg]</b> .....	5
<b>cpldeltlen[seg]</b> .....	4
<b>cpldeltba[seg]</b> .....	3
}	
}	
}	
}	
for(ch = 0; ch < nfchans; ch++)	

Syntaxe	Taille de mot
<pre> {   if(deltbae[ch]==new info follows)   {     <b>deltseg[ch]</b> .....3     for(seg = 0; seg &lt;= deltseg[ch]; seg++)     {       <b>deltfst[ch][seg]</b> .....5       <b>deltlen[ch][seg]</b> .....4       <b>deltba[ch][seg]</b> .....3     }   } } </pre>	
<pre> /* Champs pour l'inclusion de données fictives non utilisées */ <b>skiple</b> ..... 1 if(skiple) {   <b>skipl</b> .....9   <b>skipfld</b> .....skipl × 8 } </pre>	
<pre> /* Champs pour les valeurs des mantisses quantifiées */ ch = 0 do /* mantisses des canaux croissants jusqu'au premier canal couplé inclus */ {   for(bin = 0; bin &lt; nchmant[ch]; bin++) <b>{chmant[ch][bin]}</b>.....(0-16)   ch += 1 } while(chinclk[ch] == 0 &amp;&amp; ch &lt; nfchans) if(cplinu) /* mantisses du canal de couplage */ {   for(bin = 0; bin &lt; ncplmant; bin++) <b>{cplmant[bin]}</b> .....(0-16) } while(ch &lt; nfchans) /* mantisses des canaux restants, couplés ou non */ {   for(bin = 0; bin &lt; nchmant[ch]; bin++) <b>{chmant[ch][bin]}</b>.....(0-16)   ch += 1 } if(lfeon) /* mantisses du canal d'effets basse fréquence */ {   for(bin = 0; bin &lt; nlfemant; bin++) <b>{lfemant[bin]}</b> .....(0-16) } } /* fin de audblk */ </pre>	

### 5.3.4 auxdata – Données auxiliaires

Syntaxe	Taille de mot
<pre> auxdata() {   <b>auxbits</b> .....nauxbits   if(auxdatae)   {     <b>auxdata1</b> .....14   }   <b>auxdatae</b> ..... 1 } /* fin de auxdata */ </pre>	



**5.3.5 errorcheck – Code de détection d'erreur**

Syntaxe	Taille de mot
errorcheck() { <b>crcrsv</b> ..... 1 <b>crc2</b> ..... 16 } /* fin de errorcheck */	

**5.4 Description des éléments du flux binaire**

Un certain nombre d'éléments du flux binaire ont des valeurs pouvant être transmises, mais dont les significations sont réservées. Si un décodeur reçoit un flux binaire contenant des valeurs réservées, le décodeur peut ou non être capable de décoder et de produire le signal audio. La description des éléments du flux binaire ayant une valeur réservée indique ce que le décodeur peut faire quand le code réservé est reçu. Dans certains cas, le décodeur ne peut pas décoder le signal audio. Dans d'autres cas, le décodeur peut quand même décoder le signal audio en utilisant une valeur par défaut pour un paramètre qui a été indiqué par un code réservé.

**5.4.1 syncinfo – Information de synchronisation**

**5.4.1.1 syncword – Mot de synchronisation – 16 bits**

Le mot de synchronisation, **syncword**, est toujours égal à 0x0B77, ou 0000 1011 0111 0111. La transmission de **syncword** se fait avec le bit de gauche en tête, comme pour tout élément de champ binaire.

**5.4.1.2 crc1 – Contrôle de redondance cyclique 1 – 16 bits**

Ce code de CRC de 16 bits s'applique aux premiers 5/8 de la trame. La transmission du code de CRC se fait avec le bit de plus fort poids en tête, comme pour toute autre valeur numérique.

**5.4.1.3 fscod – Code de fréquence d'échantillonnage – 2 bits**

Ce code de 2 bits indique la fréquence d'échantillonnage conformément au Tableau 1. Si le code réservé est indiqué, le décodeur ne devrait pas essayer de décoder le signal audio et devrait rester muet.

TABLEAU 1

**Codes de fréquence d'échantillonnage**

fscod	Fréquence d'échantillonnage, (kHz)
00	48
01	44,1
10	32
11	Réservé

**5.4.1.4 frmsizecod – Code de longueur de trame – 6 bits**

Le code de longueur de trame est utilisé conjointement avec le code de fréquence d'échantillonnage pour déterminer le nombre de mots (de 2 octets) avant l'arrivée du mot **syncword** suivant. Voir le Tableau 13.

## 5.4.2 bsi – Information de flux binaire

### 5.4.2.1 bsid – Identification du flux binaire – 5 bits

Ce champ binaire a une valeur de 01000 (= 8) dans la présente version de la présente Norme. Des modifications futures de la présente Norme définiront peut-être d'autres valeurs. Des valeurs de bsid inférieures à 8 seront utilisées pour des versions de la Norme AC-3 qui réalisent des sous-ensembles de la syntaxe de la version 8. Des décodeurs capables de décoder la version 8 pourront donc décoder des versions de numéro inférieur à 8. Si la présente Norme est complétée par l'addition d'éléments et de fonctions supplémentaires, une valeur de bsid supérieure à 8 sera utilisée. Des décodeurs qui se conforment à la présente version de la Norme ne seront pas capables de décoder des versions possédant des identifications bsid supérieures à 8. En conséquence, des décodeurs qui se conforment à la présente Norme doivent rester muets si la valeur de bsid est supérieure à 8 et devraient décoder et reproduire les signaux audio si la valeur de bsid est inférieure ou égale à 8.

### 5.4.2.2 bsmode – Mode flux binaire – 3 bits

Ce code de 3 bits indique le type de service transporté par le flux binaire comme défini par le Tableau 2.

TABLEAU 2

#### Mode flux binaire

bsmod	acmod	Type de service
000	any	Service audio principal: service principal complet (CM)
001	any	Service audio principal: musique et effets (ME)
010	any	Service associé: malvoyant (VI)
011	any	Service associé: malentendants (HI)
100	any	Service associé: dialogue (D)
101	any	Service associé: commentaire (C)
110	any	Service associé: urgence (E)
111	'001'	Service associé: voix en surimpression (VO)
111	'010'-'111'	Service audio principal: karaoké

### 5.4.2.3 acmod – Mode codage audio – 3 bits

Ce code de trois bits, défini par le Tableau 3 indique quels sont les canaux de service principaux (entre 3/2 et 1/0) utilisés. Si le bit le plus significatif de acmod est 1, les canaux ambiophoniques sont utilisés et la suite du flux binaire contient surmixlev. Si le bit le plus significatif de acmod est 0, les canaux ambiophoniques ne sont pas utilisés et surmixlev ne suit pas. Si le bit le moins significatif de acmod est 0, le canal central n'est pas utilisé. Si le bit le moins significatif de acmod est 1, le canal central est utilisé. Il est à noter que l'état de acmod détermine le nombre nfchans de paramètres de canaux de largeur de bande nominale (par exemple, pour le mode 3/2, le nombre nfchans = 5; pour le mode 2/1, le nombre nfchans = 3; etc.). Le nombre total de canaux nchans est égal à nfchans si le canal d'effets lfe n'est pas actif (voir l'avant-propos) et égal à 1 + nfchans si le canal d'effets lfe est actif. Si acmod est nul, alors deux canaux de programme complètement indépendants (double mono) sont codés dans le flux binaire et sont référencés comme canal1 et canal2. Dans ce cas, un nombre d'articles additionnels sont présents dans BSI et audblk afin de décrire complètement le canal2. Le Tableau 3 indique l'ordre des canaux (l'ordre dans lequel les canaux sont traités) pour chacun des modes.

TABLEAU 3

**Mode codage audio**

acmod	Mode codage audio	nfchans	Ordre des canaux
000	1 + 1	2	Canal 1, canal 2
001	1/0	1	C
010	2/0	2	G, D
011	3/0	3	G, C, D
100	2/1	3	G, D, A
101	3/1	4	G, C, D, A
110	2/2	4	G, D, AG, AD
111	3/2	5	G, C, D, AG, AD

**5.4.2.4 cmixlev – Niveau du mixage central – 2 bits**

Lorsque trois canaux avant sont utilisés, ce code de 2 bits, défini par le Tableau 4, indique le niveau nominal de sous-mixage du canal central par rapport aux canaux gauche et droit. Le décodeur devrait reproduire le signal audio même si *cmixlev* prend la valeur du code réservé. La valeur intermédiaire de *cmixlev* (-4,5 dB) peut être utilisée dans ce cas.

TABLEAU 4

**Niveau du mixage central**

cmixlev	clev
00	0,707 (-3,0 dB)
01	0,596 (-4,5 dB)
10	0,500 (-6,0 dB)
11	Réservé

**5.4.2.5 surmixlev – Niveau du mixage ambiophonique – 2 bits**

Lorsque les canaux ambiophoniques sont utilisés, ce code de 2 bits défini par le Tableau 5, indique le niveau de sous-mixage des canaux ambiophoniques. Le décodeur devrait reproduire le signal audio même si *surmixlev* prend la valeur du code réservé. La valeur intermédiaire de *surmixlev* (-6 dB) peut être utilisée dans ce cas.

TABLEAU 5

**Niveau du mixage ambiophonique**

surmixlev	slev
00	0,707 (-3 dB)
01	0,500 (-6 dB)
10	0
11	Réservé

#### 5.4.2.6 dsurmod – Mode ambiophonique Dolby – 2 bits

Lors du fonctionnement en mode bi-canal, ce code de 2 bits défini par le Tableau 6, indique si le programme a été ou non codé en mode Dolby ambiophonique. Le décodeur AC-3 n'utilise pas cette information qui peut toutefois être utilisée dans d'autres parties de l'équipement de reproduction audio. Le décodeur devrait reproduire le signal audio même si dsurmod prend la valeur du code réservé. Le code réservé doit être interprété comme «pas d'indication».

TABLEAU 6  
Mode ambiophonique Dolby

dsurmod	Indication
00	Pas d'indication
01	PAS de codage ambiophonique Dolby
10	Codage ambiophonique Dolby
11	Réservé

#### 5.4.2.7 lfeon – Canal d'effets basse fréquence actif – 1 bit

Ce bit a la valeur 1 si le canal d'effets lfe (à très basse fréquence) est utilisé et la valeur 0 si le canal d'effets lfe n'est pas utilisé.

#### 5.4.2.8 dialnorm – Normalisation du dialogue – 5 bits

Ce code de 5 bits indique de combien le niveau moyen de dialogue se situe au-dessous de la valeur numérique 100%. Les valeurs acceptées vont de 1 à 31. La valeur 0 est réservée. Les valeurs de 1 à 31 sont interprétées comme –1 dB à –31 dB par rapport à la valeur numérique 100%. Si la valeur réservée 0 est reçue, le décodeur doit utiliser –31 dB. La valeur de dialnorm doit influencer le niveau de reproduction du son. Si la valeur n'est pas utilisée par le décodeur AC-3 lui-même, cette valeur doit être utilisée par une autre partie de l'équipement de reproduction audio. La normalisation du dialogue est expliquée plus en détail dans le § 7.6.

#### 5.4.2.9 compre – Présence du mot de gain de compression – 1 bit

Si ce bit est à 1, les 8 bits suivants représentent un mot de commande de la compression.

#### 5.4.2.10 compr – Mot de gain de compression – 8 bits

Ce mot de gain de compression produit par le codeur peut être présent dans le flux binaire. S'il est présent, il peut être utilisé pour ajuster le niveau audio reproduit de manière à reproduire une plage dynamique très réduite, avec une garantie de limite supérieure pour les pics instantanés du niveau de signal reproduit dans le sous-mixage monophonique. La signification et l'utilisation du mot compr sont décrits plus en détail dans le § 7.7.2.

#### 5.4.2.11 langcode – Présence du code de langue – 1 bit

Si ce bit est à 1, les 8 bits suivants représentent un code de langue. Si ce bit est à 0, la langue du service audio n'est pas indiquée.

#### 5.4.2.12 langcod – Code de langue – 8 bits

Ce code de 8 bits définit la langue du service audio. Voir le Tableau 14 pour la traduction du code langcod en langue.

#### 5.4.2.13 audprodie – Présence de l'information de production audio – 1 bit

Si ce bit est à 1, les champs mixlevel et roomtyp existent et donnent une information sur l'environnement de production audio (salle de mixage).

#### 5.4.2.14 mixlevel – Niveau du mixage – 5 bits

Ce code à 5 bits indique le niveau de pression acoustique absolu du son d'un canal individuel pendant la session de mixage audio finale. Le code à 5 bits représente une valeur comprise entre 0 et 31 dB. Le niveau de mixage maximal, qui est égal à 80 dB SPL plus la valeur de mixlevel, est compris entre 80 et 111 dB SPL. Le niveau de mixage maximal est

le niveau acoustique, dans un seul canal, d'une onde sinusoïdale dont les crêtes atteignent 100% dans la représentation MIC. D'ordinaire, la valeur SPL absolue est mesurée au moyen d'un bruit rose ayant une valeur efficace de -20 ou -30 dB par rapport au niveau efficace maximal d'une onde sinusoïdale. La valeur de `mixlevel` n'est normalement pas utilisée dans le décodeur AC-3, mais peut être utilisée par d'autres parties de l'équipement de reproduction audio.

#### 5.4.2.15 `roomtyp` – Type de salle – 2 bits

Ce code de 2 bits défini par le Tableau 7 indique le type et l'étalonnage de la salle de mixage utilisée pour la session de mixage audio finale. La valeur du code `roomtyp` n'est normalement pas utilisée dans le décodeur AC-3, mais peut être utilisée par d'autres parties de l'équipement de reproduction audio. Le décodeur devrait reproduire le signal audio même si `roomtyp` prend la valeur du code réservé. Le code réservé peut être interprété comme «pas d'indication».

TABLEAU 7

Type de salle

roomtyp	Type de salle de mixage
00	Pas d'indication
01	Grande salle, moniteur à courbe en X
10	Petite salle, moniteur à courbe plate
11	Réservé

#### 5.4.2.16 `dialnorm2` – Normalisation du dialogue, canal2 – 5 bits

Ce code de 5 bits a la même signification que `dialnorm`, mais s'applique au second canal audio lorsque `acmod` indique deux canaux indépendants (mode dual mono 1 + 1).

#### 5.4.2.17 `compr2e` – Présence du mot de gain de compression, canal2 – 1 bit

Si ce bit est à 1, les 8 bits suivants représentent un mot de gain de compression pour le canal2.

#### 5.4.2.18 `compr2` – Mot de gain de compression, canal2 – 8 bits

Ce mot de 8 bits a la même signification que `compr`, mais s'applique au second canal audio lorsque le code `acmod` indique deux canaux indépendants (mode dual mono 1 + 1).

#### 5.4.2.19 `langcod2e` – Présence du code de langue, canal2 – 1 bit

Si ce bit est à 1, les 8 bits suivants représentent un code de langue pour le canal2. Si ce bit est à 0, il n'y a pas de code de langue pour le canal2.

#### 5.4.2.20 `langcod2` – Code de langue, canal2 – 8 bits

Ce mot de 8 bits a la même signification que `langcod`, mais s'applique au second canal audio lorsque le code `acmod` indique deux canaux indépendants (mode dual mono 1 + 1).

#### 5.4.2.21 `audprodi2e` – Présence de l'information de production audio, canal2 – 1 bit

Si ce bit est à 1, les deux champs suivants existent et donnent une information sur la production audio pour le canal2.

#### 5.4.2.22 `mixlevel2` – Niveau du mixage, canal2 – 5 bits

Ce code de 5 bits a la même signification que `mixlevel`, mais s'applique au second canal audio lorsque le code `acmod` indique deux canaux indépendants (mode dual mono 1 + 1).

#### 5.4.2.23 `roomtyp2` – Type de salle, canal2 – 2 bits

Ce code de 2 bits a la même signification que `roomtyp` mais s'applique au second canal audio lorsque le code `acmod` indique deux canaux indépendants (mode dual mono 1 + 1).

**5.4.2.24 copyrightb – Bit de droit d'auteur – 1 bit**

Si ce bit a la valeur 1, il indique que l'information du flux binaire est protégée par un droit d'auteur. Il a la valeur 0 si l'information n'est pas indiquée comme étant protégée.

**5.4.2.25 origbs – Flux binaire d'origine – 1 bit**

Ce bit a la valeur 1 s'il s'agit d'un flux binaire d'origine. Ce bit a la valeur 0 s'il s'agit d'une copie d'un autre flux binaire.

**5.4.2.26 timecod1e, timecod2e – Présence de moitiés de codes temporels (premier et deuxième) – 2 bits**

Comme le montre le Tableau 8, cette valeur indique si des codes temporels suivent dans le flux binaire. Le code de temps peut avoir une résolution de 1/64 de trame (1 trame = 1/30 de seconde). Comme il n'est nécessaire de disposer que de la partie à haute résolution de code temporel pour une synchronisation fine, le code de temps à 28 bits est divisé en deux moitiés de 14 bits. La première moitié à basse résolution indique le code par incréments de 8 s jusqu'à 24 h. La deuxième moitié à haute résolution représente le code par incréments de 1/64 de trame jusqu'à 8 s.

TABLEAU 8

**Présence du code temporel**

timecod2e, timecod1e	Présence du code de temps
0,0	Non présent
0,1	Première moitié présente (14 bits)
1,0	Deuxième moitié présente (14 bits)
1,1	Deux moitiés présentes (28 bits)

**5.4.2.27 timecod1 – Première moitié du code temporel – 14 bits**

Les 5 premiers bits de ce champ de 14 bits représentent le temps en heures, avec des valeurs valides de 0 à 23. Les 6 bits suivants représentent le temps en minutes avec des valeurs valides de 0 à 59. Les 3 derniers bits représentent le temps par incréments de 8 s, avec des valeurs valides de 0 à 7 (représentant 0, 8, 16, ..., 56 s).

**5.4.2.28 timecod2 – Deuxième moitié du code temporel – 14 bits**

Les 3 premiers bits de ce champ de 14 bits représentent le temps en secondes, avec des valeurs valides de 0 à 7 (représentant 0 à 7 s). Les 5 bits suivants représentent le temps en trames avec des valeurs valides de 0 à 29. Les 6 derniers bits représentent des fractions de 1/64 de trame, avec des valeurs valides de 0 à 63.

**5.4.2.29 addbsie – Présence de l'information supplémentaire de flux binaire – 1 bit**

Si ce bit a la valeur 1, il existe une information supplémentaire de flux binaire. S'il a la valeur 0, il n'existe pas d'information supplémentaire de flux binaire.

**5.4.2.30 addbsil – Longueur de l'information supplémentaire de flux binaire – 6 bits**

Ce code de 6 bits, qui n'existe que si l'indicateur `addbsie` est à 1, indique la longueur en octets de l'information supplémentaire de flux binaire. Le domaine de valeurs valides de `addbsil` est 0 à 63, indiquant de 1 à 64 octets respectivement. Il n'est pas exigé que le décodeur interprète cette information et celui-ci peut ignorer un nombre correspondant de bits dans la suite du flux de données.

**5.4.2.31 addbsi – Information supplémentaire de flux binaire – ((addbsil + 1) × 8) bits**

Ce champ de 1 à 64 octets contient toute information additionnelle incluse dans la structure d'information du flux binaire.

### 5.4.3 audblk – Bloc audio

#### 5.4.3.1 blksw[ch] – Indicateur de commutation de bloc – 1 bit

Cet indicateur, relatif au canal [ch], indique que le bloc audio actuel a été partagé en deux sous-blocs au cours de la transformation du domaine temps vers le domaine fréquence. Une valeur de 0 indique que le bloc n'a pas été partagé et qu'une unique transformation TDAC à 512 points a été effectuée. Une valeur de 1 indique que le bloc a été partagé en deux sous-blocs de longueur 256, que la taille de la transformation TDAC a été réduite d'une longueur de 512 points à une longueur de 256 points et que deux transformations ont été effectuées sur le bloc audio (une pour chaque sous-bloc). Le changement de longueur de la transformation est présenté plus en détail dans le § 7.9.

#### 5.4.3.2 dithflag[ch] – Indicateur de bruit de désambiguïsation (dither) – 1 bit

Cet indicateur relatif, au canal [ch], indique que le décodeur devrait activer le bruit de désambiguïsation (dither) pendant le bloc en cours. Ce bruit est décrit en détail dans le § 7.3.4.

#### 5.4.3.3 dynrng – Présence du mot de plage de gain dynamique – 1 bit

Si ce bit est à 1, le mot de plage de gain dynamique suit dans le flux binaire. S'il est à 0, le mot de gain n'est pas présent et la valeur précédente est réutilisée, sauf pour le bloc 0 d'une trame pour lequel la valeur actuelle de gain dynrng est mise à 0 si le mot de commande n'est pas présent.

#### 5.4.3.4 dynrng – Mot de plage de gain dynamique – 8 bits

Ce mot de gain généré par le codeur est utilisé pour ajuster le signal audio reproduit comme décrit dans le § 7.7.1.

#### 5.4.3.5 dynrng2e – Présence du mot de plage de gain dynamique, canal2 – 1 bit

Si ce bit est à 1, le mot de plage de gain dynamique du canal2 suit dans le flux binaire. S'il est à 0, le mot de gain n'est pas présent et la valeur précédente est réutilisée, sauf pour le bloc 0 d'une trame pour lequel la valeur actuelle de gain dynrng2 est mise à 0 si le mot de commande n'est pas présent.

#### 5.4.3.6 dynrng2 – Mot de plage de gain dynamique, canal2 – 8 bits

Ce mot de gain généré par le codeur est utilisé pour ajuster le signal audio reproduit de la même façon que le gain dynrng est appliqué au canal1, comme décrit dans le § 7.7.1.

#### 5.4.3.7 cplstre – Présence de la stratégie de couplage – 1 bit

Si ce bit est à 1, une information de couplage suit dans le flux binaire. S'il est à 0, une nouvelle information de couplage n'est pas présente et les paramètres de couplage reçus précédemment sont réutilisés.

#### 5.4.3.8 cplinu – Couplage utilisé – 1 bit

Si ce paramètre est à 1, le couplage est utilisé actuellement et les paramètres de couplage suivent. S'il est à 0, le couplage n'est pas utilisé (tous les canaux sont indépendants) et aucun paramètre de couplage ne suit dans le flux binaire.

#### 5.4.3.9 chincpl[ch] – Canal couplé – 1 bit

Si ce bit est à 1, le canal indiqué par l'index [ch] est un canal couplé. Si le bit est à 0, ce canal n'est pas couplé. Comme le couplage n'est pas utilisé dans le mode 1/0, si une quelconque des valeurs chincpl[] existe, il existe de 2 à 5 telles valeurs. Parmi les valeurs présentes, deux au moins sont égales à 1 parce que le couplage exige le couplage de plus d'un canal.

#### 5.4.3.10 phsflginu – Indicateurs de phase utilisés – 1 bit

Si ce bit (défini pour le mode 2/0 seulement) est à 1, des indicateurs de phase sont inclus avec l'information de coordonnée de couplage. Les indicateurs de phase sont décrits dans le § 7.4.

#### 5.4.3.11 cplbegf – Code de fréquence de début de couplage – 4 bits

Ce code de 4 bits est interprété comme numéro de sous-bande (de 0 à 15) indiquant la limite inférieure de la bande fréquence du canal de couplage (ou la première sous-bande active) comme indiqué dans le Tableau 38.

#### 5.4.3.12 cplendf – Code de fréquence de fin de couplage – 4 bits

Ce code de 4 bits indique la limite supérieure de la bande du canal de couplage. La limite supérieure (ou la dernière sous-bande active) est égale à cplendf+2 ou à une valeur comprise entre 2 et 17. Voir le Tableau 38.

Le nombre de sous-bandes de couplage actives est égal à  $ncplsubnd$ , donné par la formule:

$$ncplsubnd = 3 + cplendf - cplbegf;$$

#### 5.4.3.13 $cplbndstrc[sbnd]$ – Structure de la bande couplage – 1 bit

Il existe 18 sous-bandes de couplage, définies dans le Tableau 38, contenant chacune 12 coefficients fréquentiels. Les sous-bandes de couplage fixes d'une largeur de 12 intervalles sont converties en bandes de couplage dont chacune peut avoir une largeur supérieure à 12 (ou un multiple de 12) intervalles de fréquence. Chaque bande couplage peut contenir une ou plusieurs sous-bandes. Les coordonnées de couplage sont transmises pour chaque bande couplage. La coordonnée de couplage de chaque bande doit être appliquée à tous les coefficients dans la bande de couplage.

La structure de la bande de couplage indique quelles sous-bandes de couplage sont combinées pour former des bandes de couplage plus larges. Lorsque  $cplbndstrc[sbnd]$  est égal à 0, la sous-bande rang  $[sbnd]$  n'est pas combinée avec la bande précédente pour former une bande plus large, mais crée une nouvelle bande couplage de largeur 12. Lorsque la valeur de  $cplbndstrc[sbnd]$  est 1, la sous-bande  $[sbnd]$  est combinée avec la bande précédente, élargissant de ce fait le bin de la bande précédente. Chaque valeur successive de  $cplbndstrc$  égale à 1 donnera lieu à une combinaison des sous-bandes avec la bande actuelle. La réception d'une nouvelle valeur de  $cplbnd$  égale à 0 donne lieu à la formation d'une nouvelle bande, commençant avec les 12 bins de la sous-bande actuelle. L'ensemble des valeurs  $cplbndstrc[sbnd]$  est considéré comme une matrice.

Chaque bit de la matrice correspond, dans l'ordre des fréquences croissantes, à une sous-bande couplage spécifique. Le premier élément de la matrice correspondant à la sous-bande  $cplbegf$  est toujours nul et n'est pas transmis. (Il n'y a aucune raison de transmettre un bit  $cplbndstrc$  pour la première sous-bande  $cplbegf$ , car ce bit est toujours égal à 0). En conséquence,  $ncplsubnd-1$  valeurs de  $cplbndstrc$  sont transmises. S'il n'y a qu'une sous-bande couplage, aucun bit  $cplbndstrc$  n'est transmis.

Le nombre de bandes de couplage  $cnplbnd$  peut être calculé à partir de  $ncplsubnd$  et  $cplnstrc$  par la formule:

$$cnplbnd = (ncplsubnd - (cplbndstrc[cplbegf+1] + \dots + cplbndstrc[cplendf+2]));$$

#### 5.4.3.14 $cplcoe[ch]$ – Présence des coordonnées de couplage – 1 bit

Les coordonnées de couplage indiquent, pour un canal donné et à l'intérieur d'une bande de couplage donnée, la fraction des coefficients de fréquence du canal de couplage devant être utilisés pour recréer tous les coefficients de fréquence du canal. Les coordonnées de couplage transmises précédemment pour ce canal sont réutilisées si de nouvelles valeurs ne sont pas reçues. On trouvera de plus amples renseignements sur le couplage au § 7.4.

La totalité des coordonnées de couplage est toujours transmise dans le bloc 0 de chaque trame de synchronisation. Si  $cplcoe[ch] = 1$ , les coordonnées de couplage du canal correspondant  $[ch]$  existent et suivent le flux binaire. Si le bit a la valeur 0, les coordonnées de couplage transmises précédemment pour ce canal sont réutilisées.

#### 5.4.3.15 $mstrcplco[ch]$ – Coordonnée principale de couplage – 2 bits

Ce paramètre associé à chaque canal fixe un facteur de gain par canal (augmentant la plage dynamique) pour les coordonnées de couplage comme indiqué dans le Tableau 9.

TABLEAU 9

Coordonnée principale de couplage

$mstrcplco[ch]$	Multiplicateur de gain $cplco[ch][bnd]$
00	1
01	$2^{-3}$
10	$2^{-6}$
11	$2^{-9}$



**5.4.3.16 cplcoexp[ch][bnd] – Exposant de la coordonnée de couplage – 4 bits**

Chaque coordonnée de couplage est constituée d'un exposant à 4 bits et d'une mantisse à 4 bits. Cet élément contient la valeur de l'exposant de la coordonnée de couplage associée au canal [ch] et à la bande [bnd]. L'index [ch] n'existe que pour les canaux couplés. L'index [bnd] prend les valeurs de 0 à ncplbnds. Voir le § 7.4.3 pour plus de détails sur l'interprétation des coordonnées de couplage.

**5.4.3.17 cplcomant[ch][bnd] – Mantisse de la coordonnée de couplage – 4 bits**

Cet élément est la mantisse à 4 bits de la coordonnée de couplage pour le canal [ch] et la bande [bnd].

**5.4.3.18 phsflg[bnd] – Indicateur de phase – 1 bit**

Cet élément, utilisé uniquement dans le mode 2/0, indique si le décodeur devrait procéder à une inversion de phase sur les mantisses du canal de couplage lorsqu'il reconstitue le signal de sortie du canal droit. L'index [bnd] peut prendre des valeurs de 0 à ncplbnd. Les indicateurs de phase sont décrits au § 7.4.

**5.4.3.19 rematstr – Stratégie de rematriçage – 1 bit**

Ce bit est à 1 lorsque de nouveaux indicateurs de rematriçage sont présents dans le flux binaire. S'il est à 0, il n'y a pas d'indicateurs de rematriçage et les valeurs précédentes devraient être réutilisées. Le paramètre rematstr est présent seulement dans le mode de codage audio 2/0.

**5.4.3.20 rematflg[sbnd] – Indicateur de rematriçage – 1 bit**

Ce bit indique si les coefficients de transformation dans la bande de rematriçage [rbnd]. Si ce bit est à 1, les coefficients de transformation dans [rbnd] ont subi un rematriçage donnant des canaux de somme et de différence. Si ce bit est à 0, le rematriçage n'a pas eu lieu pour la bande [rbnd]. Le nombre de bandes avec rematriçage et le nombre de valeurs de [rbnd] est fonction des paramètres de couplage comme l'indique le Tableau 10. Le rematriçage est décrit dans le § 7.5.

TABLEAU 10

Nombre de bandes avec rematriçage

Condition	Nombre de bandes avec rematriçage
cplinu == 0	4
(cplinu == 1) && (cplbegf > 2)	4
(cplinu == 1) && (2 ≥ cplbegf > 0)	3
(cplinu == 1) && (cplbegf == 0)	2

**5.4.3.21 cplexpstr – Stratégie d'exposant de couplage – 2 bits**

Cet élément indique la méthode de codage d'exposant utilisée pour le canal de couplage, comme indiqué dans le Tableau 18. Voir le § 7.1 pour des explications concernant les stratégies d'exposant.

**5.4.3.22 chexpstr[ch] – Stratégie d'exposant de canal – 2 bits**

Cet élément indique la méthode de codage d'exposant utilisée pour le canal [ch], comme indiqué dans le Tableau 18. Cet élément existe pour tout canal de largeur de bande complète.

**5.4.3.23 lfeexpstr – Stratégie d'exposant de canal d'effets basse fréquence – 1 bit**

Cet élément indique la méthode de codage d'exposant utilisée pour le canal d'effets lfe, comme indiqué dans le Tableau 19.

**5.4.3.24 chbwcod[ch] – Code de largeur de bande canal – 6 bits**

L'élément chbwcod[ch] est un entier sans signe qui définit la limite supérieure de la bande d'un canal [ch] de largeur de bande complète. Ce paramètre n'est inclus que pour les canaux de largeur de bande complète qui ne sont pas couplés. (Voir le § 7.1.3 en ce qui concerne les exposants pour la définition de ce paramètre.) Les valeurs valides appartiennent au domaine de 0 à 60. Si une valeur supérieure à 60 est reçue, le flux binaire n'est pas valide et le décodeur devrait arrêter le décodage et rester muet.

**5.4.3.25 cplabsexp – Exposant absolu de couplage – 4 bits**

Ceci est un exposant absolu utilisé comme référence pour le décodage des exposants différentiels pour le canal de couplage.

**5.4.3.26 cplexps[grp] – Exposants de couplage – 7 bits**

Chaque valeur de cplexps indique, pour le groupe [grp] d'exposants du canal de couplage, la valeur de 3, 6 ou 12 exposants du canal de couplage codés de manière différentielle respectivement avec les stratégies d'exposant D15, D25 ou D45. Le nombre de valeurs de cplexps transmises est égal à ncplgrps, qui peut être calculé à partir de cplbegf, cplendf et cplexpstr. Voir le § 7.1.3 pour de plus amples détails.

**5.4.3.27 exps[ch][grp] – Exposants de canal – 4 ou 7 bits**

Ces éléments représentent les exposants codés pour le canal [ch]. Le premier élément ([grp] = 0) est l'exposant absolu à 4 bits pour le premier coefficient de la transformée (terme continu). Les éléments suivants ([grp] > 0) sont des représentations à 7 bits d'un groupe de 3, 6 ou 12 exposants codés de manière différentielle, pour les stratégies d'exposant D15, D25 et D45 respectivement. Le nombre de groupes pour chaque canal, nchgrps[ch] est déterminé à partir de cplbegf si le canal est couplé ou à partir de nchbwcod[ch] si le canal n'est pas couplé. Voir le § 7.1.3 pour plus d'informations.

**5.4.3.28 gainrng[ch] – Code de plage de gain de canal – 2 bits**

Cet élément de 2 bits par canal peut être utilisé pour déterminer une valeur de décalage de bloc en virgule flottante pour le banc de filtres de la transformation TDAC inverse. L'utilisation de ce paramètre permet d'obtenir une plage dynamique accrue à partir d'un calcul de transformée utilisant une taille de mot limitée. Voir le § 7.9.5 pour plus d'informations.

**5.4.3.29 lfeexps[grp] – Exposants de canal d'effets basse fréquence – 4 ou 7 bits**

Ces éléments représentent les exposants codés pour le canal d'effets lfe. Le premier élément ([grp] = 0) est l'exposant absolu à 4 bits du premier coefficient de la transformée (terme continu). Il existe deux autres éléments (nlfegrps = 2) qui sont des représentations à 7 bits d'un groupe de trois exposants codés d'une manière différentielle. Le nombre total d'exposants du canal d'effets lfe (nlfemant) est de 7.

**5.4.3.30 baie – Présence de l'information d'affectation de bits – 1 bit**

Si ce bit est à 1, cinq champs distincts (totalisant 11 bits) suivent dans le flux binaire. Chaque champ indique des valeurs de paramètres pour le processus d'affectation de bits. Si ce bit est à 0, ces champs n'existent pas. Des détails supplémentaires peuvent être trouvés dans le § 7.2.

**5.4.3.31 sdcycod – Code de décroissance lente – 2 bits**

Ce code de 2 bits spécifie le paramètre de décroissance lente du processus d'affectation de bits.

**5.4.3.32 fdcycod – Code de décroissance rapide – 2 bits**

Ce code de 2 bits spécifie le paramètre de décroissance rapide du processus d'affectation de bits.

**5.4.3.33 sgaincod – Code de gain lent – 2 bits**

Ce code de 2 bits spécifie le paramètre de gain lent du processus d'affectation de bits du décodage.

**5.4.3.34 dbpbcod – Code de dB par bit – 2 bits**

Ce code de 2 bits spécifie le paramètre dB par bit du processus d'affectation de bits.

**5.4.3.35 floorcod – Code de plancher de masquage – 3 bits**

Ce code de 2 bits spécifie le paramètre du code de plancher du processus d'affectation de bits.

**5.4.3.36 snroffste – Présence du déplacement du rapport S/N – 1 bit**

Si ce bit a la valeur 1, un certain nombre de paramètres d'affectation de bits suivent dans le flux binaire. Si ce bit a la valeur 0, l'information de déplacement du rapport S/N ne suit pas et les valeurs transmises précédemment devraient être utilisées pour ce bloc. Le processus d'affectation de bits et ces paramètres sont décrits au § 7.2.2.

**5.4.3.37 csnroffst – Déplacement du rapport S/N approximatif – 6 bits**

Ce code de 6 bits spécifie le déplacement du paramètre de rapport S/N approximatif dans le processus d'affectation de bits.

**5.4.3.38 cplfsnroffst – Déplacement fin du rapport S/N du couplage – 4 bits**

Ce code de 4 bits spécifie le déplacement du rapport S/N fin du canal de couplage dans le processus d'affectation de bits.

**5.4.3.39 cplfgaincod – Code de couplage du gain rapide – 3 bits**

Ce code de 3 bits spécifie le code de gain rapide du canal de couplage utilisé dans le processus d'affectation de bits.

**5.4.3.40 fsnroffst[ch] – Déplacement fin du rapport S/N du canal – 4 bits**

Ce code de 4 bits spécifie le déplacement fin du rapport S/N utilisé dans le processus d'affectation de bits pour le canal [ch].

**5.4.3.41 fgaincod[ch] – Code de gain rapide du canal – 3 bits**

Ce code de 3 bits spécifie le paramètre de gain rapide utilisé dans le processus d'affectation de bits pour le canal [ch].

**5.4.3.42 lfefsnroffst – Déplacement fin du rapport S/N du canal d'effets lfe – 4 bits**

Ce code de 4 bits spécifie le déplacement fin du rapport S/N utilisé dans le processus d'affectation de bits pour le canal d'effets lfe.

**5.4.3.43 lfefgaincod – Code de gain rapide du canal d'effets basse fréquence – 3 bits**

Ce code de 3 bits spécifie le paramètre de gain rapide utilisé dans le processus d'affectation de bits pour le canal d'effets lfe.

**5.4.3.44 cplleake – Présence de l'initialisation de fuite de couplage – 1 bit**

Si ce bit est à 1, des paramètres d'initialisation de fuite du couplage suivent dans le flux binaire. Si ce bit est à 0, les valeurs transmises précédemment continuent à s'appliquer.

**5.4.3.45 cplfleak – Initialisation de fuite rapide couplage – 3 bits**

Ce code de 3 bits spécifie la valeur de l'initialisation de la fuite rapide pour le calcul de la fonction d'excitation du canal de couplage dans le processus d'affectation de bits.

**5.4.3.46 cplleak – Initialisation de fuite lente de couplage – 3 bits**

Ce code de 3 bits spécifie la valeur de l'initialisation de la fuite lente pour le calcul de la fonction d'excitation du canal de couplage dans le processus d'affectation de bits.

**5.4.3.47 deltbaie – Présence de l'information d'affectation de bits delta – 1 bit**

Si ce bit est à 1, une certaine information d'affectation de bits delta suit dans le flux binaire. Si ce bit est à 0, l'information d'affectation de bits delta transmise précédemment continue à s'appliquer, sauf pour le bloc 0. Si deltbaie est égal à 0 pour le bloc 0, alors cpldeltseg et deltnseg[ch] sont mis à 0 et aucune affectation de bits delta ne s'applique. L'affectation de bits delta est décrite dans le § 7.2.2.6.

**5.4.3.48 cpldeltbae – Présence de l'affectation de bits delta pour le canal de couplage – 2 bits**

Ce code de 2 bits indique la stratégie d'affectation de bits delta pour le canal de couplage, comme définie dans le Tableau 11. Si l'état réservé est reçu, le décodeur ne devrait pas décoder le signal audio et devrait rester muet.

**5.4.3.49 deltbae[ch] – Présence de l'affectation de bits delta – 2 bits**

Ce code de 2 bits indique la stratégie d'affectation de bits delta pour chaque canal de largeur de bande complète, telle que définie dans le Tableau 11.

**5.4.3.50 cpldeltnseg – Nombre de segments de l'affectation de bits delta pour le canal de couplage – 3 bits**

Ce code de 3 bits indique le nombre de segments d'affectation de bits delta existant pour le canal de couplage. La valeur de ce paramètre, de 1 à 8, est calculée en ajoutant 1 au nombre binaire à 3 bits représenté par ce code.

TABLEAU 11

## Etats associés à la présence de l'affectation de bits delta

cpdeltbae, deltbae	Code
00	Réutilisation de l'état précédent
01	Une nouvelle information suit
10	Pas d'affectation delta
11	Réservé

**5.4.3.51 cpdeltfst[seg] – Déplacement de l'affectation de bits delta pour le canal de couplage – 5 bits**

Le premier code 5 bits ([seg] = 0) indique le rang de la première bande d'affectation de bits (comme spécifiée au § 7.4.2) du canal de couplage pour lequel les valeurs d'affectation des bits delta sont fournies. Les codes subséquents indiquent le déplacement entre la fin du segment delta précédent et la bande d'affectation de bits suivante pour laquelle les valeurs d'affectation de bits delta sont fournies.

**5.4.3.52 cpdeltlen[seg] – Longueur de l'affectation de bits delta pour le canal de couplage – 4 bits**

Chaque code à 4 bits indique le nombre de bandes d'affectation de bits couverte par le segment correspondant.

**5.4.3.53 cpdeltba[seg] – Affectation de bits delta pour le canal de couplage – 3 bits**

Cette valeur à 3 bits est utilisée dans le processus d'affectation de bits pour le canal de couplage.

Chaque code à 3 bits indique un ajustement par rapport à la courbe de masquage par défaut calculée par le décodeur. Le codage des bits delta est indiqué dans le Tableau 12.

TABLEAU 12

## Codage des bits delta d'affectation de bits

cpdeltba, deltba	Ajustement (dB)
000	-24
001	-18
010	-12
011	-6
100	+6
101	+12
110	+18
111	+24

**5.4.3.54 deltnseg[ch] – Nombre de segments de l'affectation de bits delta d'un canal – 3 bits**

Ces éléments sont des codes à 3 bits indiquant, pour chaque canal de largeur de bande complète, le nombre de segments d'affectation de bits delta existant pour le canal correspondant. La valeur de ce paramètre est comprise entre 1 et 8 et calculée en ajoutant 1 au code binaire à 3 bits.

**5.4.3.55 deltoffst[ch][seg] – Déplacement de l'affectation de bits delta d'un canal – 5 bits**

Le premier code de 5 bits ([seg] = 0) indique le numéro de la première bande d'affectation de bits (voir le § 7.2.2.6) du canal correspondant pour lequel les valeurs d'affectation des bits delta sont fournies. Les codes subséquents indiquent le déplacement entre la fin du segment delta précédent et la bande d'affectation de bits suivante pour laquelle les valeurs d'affectation de bits delta sont fournies.

**5.4.3.56 delten[ch][seg] – Longueur de l'affectation de bits delta d'un canal – 4 bits**

Chaque code de 4 bits indique le nombre de bandes d'affectation de bits couvertes par le segment correspondant.

**5.4.3.57 deltba[ch][seg] – Affectation de bits delta d'un canal – 3 bits**

Cette valeur à 3 bits est utilisée dans le processus d'affectation de bits pour le canal indiqué. Chaque code de 3 bits indique un ajustement par rapport à la courbe de masquage par défaut calculé par le décodeur. Le codage des bits delta est indiqué dans le Tableau 12.

**5.4.3.58 skiple – Présence de la longueur de saut – 1 bit**

Si ce bit est à 1, le paramètre skipl suit dans le flux binaire. Si ce bit est 0, skipl n'est pas présent.

**5.4.3.59 skipl – Longueur de saut – 9 bits**

Ce code de 9 bits indique le nombre de bits sans signification à sauter (à ignorer) avant de dépaqueter les mantisses du bloc audio courant.

**5.4.3.60 skipfld – Champ sauté – (skipl × 8) bits**

Ce champ contient les octets vides à sauter, comme indiqué par le paramètre skipl.

**5.4.3.61 chmant[ch][bin] – Mantisses de canal – 0 à 16 bits**

Les valeurs effectives des mantisses quantifiées pour le canal indiqué. Chaque valeur peut contenir de 0 à 16 bits. Le nombre de mantisses pour le canal indiqué est égal à nchmant[ch], qui peut être déterminé à partir de chbwcod[ch] (voir le § 7.1.3) si le canal n'est pas couplé, et à partir de cplbegf (voir le § 7.4.2) si le canal est couplé. Voir le § 7.3 pour de plus amples détails sur les données de mantisses empaquetées.

**5.4.3.62 cplmant[bin] – Mantisses du canal de couplage – 0 à 16 bits**

Les valeurs effectives des mantisses quantifiées pour le canal de couplage. Chaque valeur peut contenir de 0 à 16 bits. Le nombre de mantisses pour le canal de couplage est égal à ncplmant, qui peut être déterminé par la formule:

$$\text{cplmant} = 12 \times \text{ncplsubnd};$$

**5.4.3.63 lfemant[bin] – Mantisses de canal d'effets basse fréquence – 0 à 16 bits**

Les valeurs effectives des mantisses quantifiées pour le canal d'effets lfe. Chaque valeur peut contenir de 0 à 16 bits. La valeur de nlfemant est égale à 7, de sorte qu'il existe 7 valeurs de mantisse pour le canal d'effets lfe.

**5.4.4 auxdata – Champ de données auxiliaires**

Des données inutilisées se trouvent en fin de trame lorsque le codeur n'utilise pas toutes les données disponibles pour coder le signal audio. Ce cas se présente lorsque la dernière affectation de bits n'arrive pas à utiliser tous les bits disponibles, ou tout simplement si le signal audio entrant n'impose pas le codage transparent de tous les bits disponibles. Le codeur peut également avoir reçu l'ordre de laisser intentionnellement quelques bits inutilisés par codage audio de manière à ce qu'ils soient disponibles pour les données auxiliaires. Comme le nombre de bits utilisés par les données auxiliaires peut être inférieur au nombre de bits disponibles (qui peut varier avec le temps) dans une trame donnée, on dispose d'une méthode pour indiquer le nombre de bits de données auxiliaires réelles dans chaque trame.

**5.4.4.1 auxbits – Bits de données auxiliaires – nauxbits bits**

Ce champ contient des données auxiliaires. Le nombre total de bits de ce champ est donné par la formule:

$$\text{nauxbits} = (\text{bits de la trame}) - (\text{bits utilisés par tous les éléments du flux binaire sauf pour auxbits});$$

Le nombre de bits de la trame peut être déterminé à partir du code de longueur de trame (frmsizcod) et du Tableau 13. Le nombre de bits utilisés inclut tous les bits utilisés par les éléments du flux binaire à l'exception des bits de données auxiliaires. Toutes les données fictives qui ont été incluses avec des champs sauts (skipfld) font partie du décompte des bits. La longueur du champ auxbits est ajustée par le codeur de manière à ce que l'élément crc2 coïncide avec le dernier mot de 16 bits de la trame.

TABLEAU 13

Code de longueur de trame (1 mot = 16 bits)

frmsizecod	Taux binaire nominal (kbit/s)	Mots/syncframe $f_s = 32$ kHz	Mots/syncframe $f_s = 44,1$ kHz	Mots/syncframe $f_s = 48$ kHz
000000 (0)	32	96	69	64
000001 (0)	32	96	70	64
000010 (1)	40	120	87	80
000011 (1)	40	120	88	80
000100 (2)	48	144	104	96
000101 (2)	48	144	105	96
000110 (3)	56	168	121	112
000111 (3)	56	168	122	112
001000 (4)	64	192	139	128
001001 (4)	64	192	140	128
001010 (5)	80	240	174	160
001011 (5)	80	240	175	160
001100 (6)	96	288	208	192
001101 (6)	96	288	209	192
001110 (7)	112	336	243	224
001111 (7)	112	336	244	224
010000 (8)	128	384	278	256
010001 (8)	128	384	279	256
010010 (9)	160	480	348	320
010011 (9)	160	480	349	320
010100 (10)	192	576	417	384
010101 (10)	192	576	418	384
010110 (11)	224	672	487	448
010111 (11)	224	672	488	448
011000 (12)	256	768	557	512
011001 (12)	256	768	558	512
011010 (13)	320	960	696	640
011011 (13)	320	960	697	640
011100 (14)	384	1152	835	768
011101 (14)	384	1152	836	768
011110 (15)	448	1344	975	896
011111 (15)	448	1344	976	896
100000 (16)	512	1536	1114	1024
100001 (16)	512	1536	1115	1024
100010 (17)	576	1728	1253	1152
100011 (17)	576	1728	1254	1152
100100 (18)	640	1920	1393	1280
100101 (18)	640	1920	1394	1280

 $f_s$ : fréquence d'échantillonnage

Si le nombre de bits d'utilisateur indiqué par `auxdata1` est inférieur au nombre de bits `auxbits` `nauxbits`, les données d'utilisateur sont localisées à la fin du champ `auxbits`. Ceci permet à un décodeur de trouver et de dépaqueter les bits d'utilisateur `auxdata1` sans connaître la valeur de `nauxbits` (qui peut seulement être déterminée en décodant les données audio de la trame entière). L'ordre des données utilisateur dans le champ `auxbits` correspond à l'ordre d'arrivée. En conséquence, le décodeur de données auxiliaires (qui peut ne décoder aucun signal audio) peut simplement rechercher `auxdata1` à la fin de la trame de synchronisation AC-3, revenir en arrière dans le flux binaire d'un nombre `auxdata1` de bits depuis le début du champ `auxdata1` puis dépaqueter `auxdata1` de bits dans l'ordre d'arrivée du flux binaire.

#### 5.4.4.2 `auxdata1` – Longueur des données auxiliaires – 14 bits

Cette valeur entière indique la longueur en bits des données utilisateur dans le champ auxiliaire `auxbits`.

#### 5.4.4.3 `auxdatae` – Présence de données auxiliaires – 1 bit

Si ce bit est à 1, alors le paramètre `auxdata1` précède dans le flux binaire. Si ce bit est à 0, `auxdata1` n'existe pas et il n'y a pas de données utilisateur.

#### 5.4.5 `errorcheck` – Champ de détection d'erreur de trame

##### 5.4.5.1 `crcsv` – Bit réservé au CRC – 1 bit

Réservé pour une utilisation dans des applications spécifiques pour garantir que `crc2` n'est pas identique au mot de synchronisation. L'utilisation de ce bit par le codeur est facultative. Si la valeur du `crc2` calculée est égale à `syncword`, le bit `crcsv` peut être interverti. Dans ce cas, le `crc2` diffère de `syncword`.

##### 5.4.5.2 `crc2` – Contrôle de redondance cyclique 2 – 16 bits

Le CRC à 16 bits s'applique à la trame entière. Les détails de vérification du CRC sont donnés au § 7.10.1.

### 5.5 Contraintes de flux binaire

Le codeur AC-3 impose les contraintes suivantes au flux binaire codé. Ces contraintes permettent de réaliser des décodeurs AC-3 dotés de mémoires tampon en entrée plus réduites.

- Les tailles combinées du bloc 0 et du bloc 1 ne dépassent jamais les 5/8 de la trame.
- La somme des données de mantisse et des données auxiliaires du bloc 5 ne dépasse jamais les derniers 3/8 de la trame.
- Le bloc 0 contient toujours toute l'information nécessaire pour commencer correctement le décodage du flux binaire.
- Chaque fois que l'état de `cplinu` passe d'inactif à actif, toute l'information de couplage nécessaire est incluse dans le bloc dans lequel le couplage est activé. Aucune information portant sur le couplage n'est réutilisée à partir d'un quelconque bloc antérieur pour lequel le couplage aurait pu être actif.

## 6 Décodage du flux binaire AC-3

### 6.1 Introduction

L'avant-propos de la présente Norme spécifie les détails de la syntaxe du flux binaire AC-3. Le présent paragraphe donne une description générale du processus de décodage AC-3 illustré par le diagramme de la Fig. 13. Dans ce diagramme, le processus de décodage est présenté sous la forme d'une séquence de blocs verticaux, certains flux d'information étant indiqués par des flèches sur les côtés de la page. Le § 7 donne de plus amples détails sur certains des blocs de traitement. Le décodeur présenté dans le présent paragraphe devrait être considéré comme un exemple de décodeur. D'autres techniques de réalisation de décodeurs existent, ces autres techniques peuvent présenter certains avantages techniques, concernant le nombre d'instructions, les besoins en mémoire, le nombre de transformations nécessaires, etc.

TABLEAU 14

## Codes de langue

langcod	Langue	langcod	Langue	langcod	Langue	langcod	Langue
00	Inconnu/ non applicable	20	Polonais	40	Son d'arrière plan/	60	Moldave
01	Albanais	21	Portugais	41		61	Malais
02	Breton	22	Roumain	42		62	Malgache
03	Catalan	23	Romain	43		63	Macédonien
04	Croate	24	Serbe	44		64	Laotien
05	Gallois	25	Slovaque	45	Zoulou	65	Coréen
06	Tchèque	26	Slovène	46	Vietnamien	66	Khmer
07	Danois	27	Finnois	47	Ouzbèque	67	Kazakh
08	Allemand	28	Suédois	48	Ourdou	68	Canada
09	Anglais	29	Turc	49	Ukrainien	69	Japonais
0A	Espagnol	2A	Flamand	4A	Thailandais	6A	Indonésien
0B	Espéranto	2B	Wallon	4B	Télougou	6B	Hindi
0C	Estonien	2C		4C	Tatare	6C	Hébreu
0D	Basque	2D		4D	Tamile	6D	Haoussa
0E	Féroïen	2E		4E	Tadjik	6E	Guarani
0F	Français	2F		4F	Swahili	6F	Goujrati
10	Frision	30	Réservé pour une utilisation nationale	50	Tongo	70	Grec
11	Irlandais	31		51	Somali	71	Géorgien
12	Gaélique	32		52	Cinghalais	72	Foulani
13	Galicien	33		53	Shona	73	Dari
14	Islandais	34		54	Serbo-croate	74	Chourache
15	Italien	35		55	Ruthène	75	Chinois
16	Japon	36		56	Russe	76	Birman
17	Latin	37		57	Quichua	77	Bulgare
18	Letton	38		58	Pustu	78	Bengali
19	Luxembourgeois	39		59	Pendjabi	79	Biélorusse
1A	Lithuanien	3A		5A	Persan	7A	Bambora
1B	Hongrois	3B		5B	Papamiento	7B	Azerbaïdjanais
1C	Maltais	3C		5C	Auriya	7C	Assamais
1D	Hollandais	3D		5D	Népalais	7D	Arménien
1E	Norvégien	3E		5E	Ndebele	7E	Arabe
1F	Occitan	3F		5F	Marathe	7F	Amharique

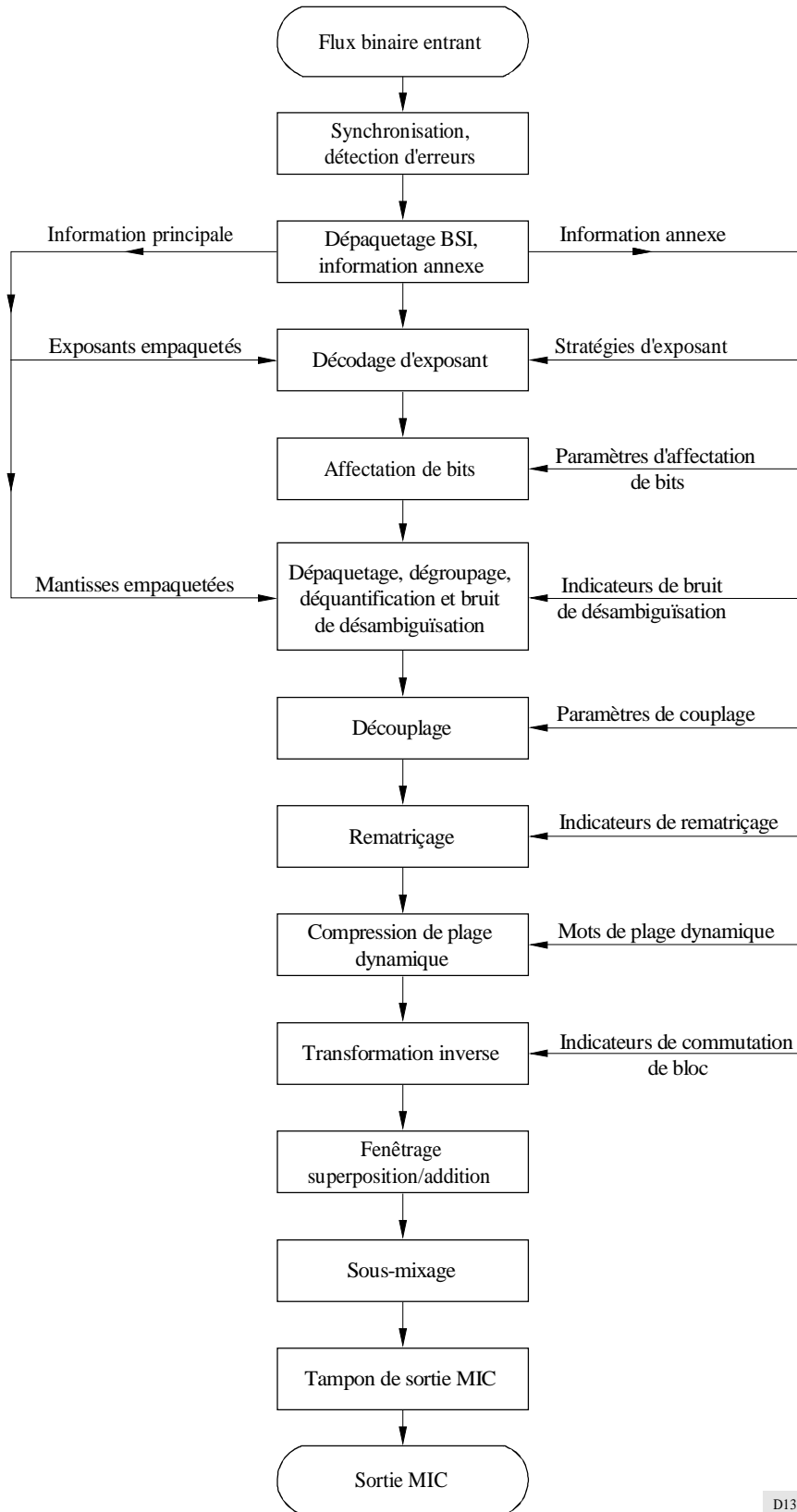
## 6.2 Description générale du processus de décodage

### 6.2.1 Flux binaire entrant

Le flux binaire entrant provient normalement d'un système de transmission ou de stockage. L'interface entre la source des données AC-3 et le décodeur AC-3 n'est pas spécifiée par la présente Norme. Les détails de cet interface influencent de nombreux détails de mise en œuvre.



FIGURE 13  
Diagramme de fluence du processus de décodage



### 6.2.1.1 Arrivée continue ou par rafales

Les données AC-3 codées peuvent être présentées au décodeur sous la forme d'un flux continu de données au débit binaire nominal, ou des tranches de données peuvent arriver au décodeur par rafales avec un débit binaire élevé et un faible facteur d'utilisation. En mode rafale, l'arrivée des rafales peut être commandée par la source de données ou le décodeur. Le tampon d'entrée du décodeur AC-3 peut être plus réduit si celui-ci peut demander les rafales de données selon ses besoins. Toutefois, dans ce cas la taille de la mémoire tampon externe peut être plus élevée.

### 6.2.1.2 Cadrage à l'octet ou au mot

La plupart des mises en œuvre de réalisations conformes à la présente Norme permettront l'acheminement du flux binaire AC-3 élémentaire avec un cadrage sur des octets ou des mots de 16 bits. La longueur de la trame de synchronisation est toujours un nombre entier de mots. Le décodeur peut recevoir les données sous la forme d'un flux continu sériel de bits sans aucun cadrage. Les données peuvent également être envoyées au décodeur avec cadrage sur des octets ou des mots de 16 bits. Le cadrage sur des octets ou sur des mots peut permettre une certaine simplification du décodeur. Le cadrage ne diminue pas la probabilité de fausse détection du mot de synchronisation.

### 6.2.2 Synchronisation et détection d'erreurs

Le format du flux binaire AC-3 permet une synchronisation rapide. La probabilité de fausse détection du mot de synchronisation de 16 bits est faible. Sans cadrage du flux binaire entrant, la probabilité de fausse détection du mot de synchronisation est de 0,0015% par position de bit du flux binaire entrant. Pour un débit binaire de 384 kbit/s, la probabilité de fausse détection erronée du mot de synchronisation est de 19% par trame. Un cadrage du flux entrant sur les octets réduit cette probabilité à 2,5% et un cadrage sur les mots l'abaisse à 1,2%.

Lorsqu'une séquence de synchronisation est détectée, on peut supposer que le décodeur est synchronisé et alors l'un des mots de CRC (`crc1` ou `crc2`) peut être vérifié. Comme le mot `crc1` apparaît en premier et couvre les 5/8 de la trame, le résultat de la vérification du mot `crc1` peut être disponible après que seulement les 5/8 de la trame ont été reçus. Il est également possible de vérifier le mot `crc2` après réception de la totalité de la trame. Si l'un ou l'autre des mots CRC est correct, on peut tout à fait estimer que le décodeur est synchronisé et qu'il est possible de procéder au décodage et à la reproduction audio. La probabilité de fausse synchronisation sera dans ce cas égale à la probabilité combinée de détection erronée du mot de synchronisation et de fausse détection d'erreur par le CRC. La vérification du CRC a une probabilité d'erreur de 0,0015%. La combinaison de cette probabilité avec la probabilité de détection erronée du mot de synchronisation fournit, pour un flux binaire entrant aligné sur les octets, une probabilité de synchronisation erronée de 0,000035% (soit environ une erreur pour 3 millions de tentatives de synchronisation).

Si cette faible probabilité de synchronisation erronée est encore trop élevée pour une application, il existe plusieurs moyens pour l'abaisser. Le décodeur peut ne faire l'hypothèse d'une synchronisation correcte que si les deux mots CRC sont les bons. Le décodeur peut exiger de recevoir plusieurs mots de synchronisation avec le cadrage approprié. Si le système de transmission ou de stockage de données a connaissance d'erreur dans les données, cette information peut être communiquée au décodeur.

La présente Norme ne donne aucun autre détail sur les méthodes de synchronisation du flux binaire. Des détails sur le calcul de CRC sont donnés dans le § 7.10.

### 6.2.3 Dépaquetage de l'information du flux binaire, information annexe

Le dépaquetage ou démultiplexage des divers types d'information contenus dans le flux binaire est inhérent au processus de décodage. Certaines de ces entités peuvent être copiées depuis le tampon d'entrée vers des registres spécialisés, d'autres peuvent être copiées dans des zones spécifiques de la mémoire de travail et d'autres encore peuvent être simplement placées dans le tampon d'entrée avec des pointeurs pointant sur elles et sauvegardées dans une autre adresse pour être utilisées en cas de besoin. L'information devant être dépaquetée est spécifiée en détail au § 5.3. La présente Norme ne donne pas d'autres détails relatifs au dépaquetage de l'information du flux binaire et de l'information annexe.

### 6.2.4 Décodage des exposants

Les exposants sont fournis dans le flux binaire sous forme codée. Deux types d'information annexe sont nécessaires pour dépaqueter et décoder les exposants. Il faut tout d'abord connaître le nombre d'exposants. Pour les canaux de largeur de bande complète, ce nombre peut être déterminé à partir des codes `chbwcod[ch]` pour les canaux non couplés ou du code `cplbegf` pour les canaux couplés. Pour le canal de couplage, le nombre d'exposants peut être déterminé à partir des codes `cplbegf` et `cplendf`. Pour le canal d'effets basse fréquence `lfe`, s'il est activé, le nombre d'exposants est toujours égal à 7. Il faut en second lieu connaître la stratégie d'exposant (`D15`, etc.) utilisée par chaque canal. Les détails concernant la manière d'empaqueter et de dépaqueter les exposants sont donnés au § 7.1.

### 6.2.5 Affectation de bits

Le calcul d'affectation de bits donne le nombre de bits utilisés par chaque mantisse. Les données d'entrée du calcul d'affectation de bits sont les exposants décodés et l'information annexe d'affectation de bits. Le résultat du calcul d'affectation de bits est un ensemble de pointeurs d'affectation de bits (baps) sous la forme d'un bap pour chaque mantisse codée. Le bap indique le quantificateur utilisé pour la mantisse et le nombre de bits occupés par la mantisse dans le flux binaire. Le calcul d'affectation de bits est décrit en détail au § 7.2.

### 6.2.6 Traitement des mantisses

L'essentiel du flux binaire AC-3 est constitué de mantisses quantifiées approximativement. Chaque mantisse est quantifiée avec le niveau de précision indiqué par le bap correspondant. Certaines mantisses sont groupées en une unique valeur transmise afin de les empaqueter d'une manière plus efficace. Par exemple, deux valeurs quantifiées au niveau 11 sont véhiculées dans un code unique de 7 bits (3,5 bits/valeur) dans le flux binaire.

La donnée de mantisse est dépaquetée en détachant des groupes de bits selon les indications fournies par les baps. Les mantisses groupées doivent être dégroupées. Les valeurs individuelles de mantisse codée sont converties en valeurs déquantifiées. Les mantisses indiquées comme ayant une valeur nulle peuvent être reproduites soit comme des zéros, soit par une valeur aléatoire de bruit de désambiguïsation (dither) (sous le contrôle de l'indicateur de bruit de désambiguïsation (dither)). Le traitement des mantisses est décrit dans le détail au § 7.3.

### 6.2.7 Découplage

Lorsque le couplage est utilisé, les canaux qui sont couplés doivent être découplés. Le découplage implique la reconstitution de la partie à haute fréquence (exposants et mantisses) de chaque canal couplé à partir du canal commun de couplage et des coordonnées de couplage du canal individuel. A l'intérieur de chaque bande couplage, il faut multiplier les coefficients du canal de couplage (exposants et mantisses) par les coordonnées de couplage de chaque canal. Le processus de couplage est décrit dans le détail au § 7.4.

### 6.2.8 Rematriçage

Le rematriçage peut être utilisé dans le mode de codage audio 2/0, comme indiqué par les indicateurs de rematriçage (`rematflg[rband]`). Lorsque l'indicateur indique qu'une bande est rematriçée, les coefficients codés dans le flux binaire sont les valeurs des sommes et des différences au lieu des valeurs gauches et droites. Le rematriçage est décrit dans le détail au § 7.5.

### 6.2.9 Compression de plage dynamique

Une valeur de commande de la plage dynamique (`dynrng`) peut être incluse dans le flux binaire. Le décodeur doit utiliser par défaut cette valeur pour modifier l'amplitude du coefficient (exposant et mantisse) comme spécifié au § 7.7.1.

### 6.2.10 Transformation inverse

Les étapes de décodage décrites ci-dessus permettent d'obtenir un ensemble de coefficients de fréquence pour chaque canal codé. La transformation inverse convertit les blocs de fréquence en blocs d'échantillons temporels. La transformation inverse est décrite dans le détail au § 7.9.

### 6.2.11 Fenêtre, superposition/addition

Chaque bloc d'échantillons temporels doit subir un fenêtrage et les blocs adjacents doivent être superposés partiellement et additionnés afin de reconstituer un signal MIC audio continu dans le temps. La fenêtre et les étapes de superposition et d'addition sont décrites en même temps que la transformation inverse au § 7.9.

### 6.2.12 Sous-mixage

Un sous-mixage est nécessaire si le nombre de canaux exigés à la sortie du décodeur est inférieur au nombre de canaux codés dans le flux binaire. Un sous-mixage dans le domaine temps est effectué dans le décodeur présenté ici à titre d'exemple. Comme la transformation inverse est une opération linéaire, il est également possible d'effectuer un sous-mixage avant la transformation. Le § 7.8 décrit le sous-mixage et spécifie les coefficients de sous-mixage que les décodeurs doivent utiliser.

### 6.2.13 Tampon de sortie MIC

Les décodeurs délivrent normalement les échantillons de sortie MIC à la fréquence d'échantillonnage MIC. Comme les blocs d'échantillons sont le résultat du processus de décodage, une mémoire tampon est normalement nécessaire en sortie. La présente Norme ne spécifie ou ne décrit pas avec plus de détail la mémorisation dans le tampon de sortie.

## 6.2.14 Sortie MIC

Les échantillons MIC de sortie peuvent être fournis sous une forme adaptée à une interconnexion avec un convertisseur numérique/analogique ou sous toute autre forme. La présente Norme ne spécifie pas le format de sortie MIC.

# 7 Description détaillée de l'algorithme

Les paragraphes qui suivent décrivent en détail divers aspects du codage AC-3.

## 7.1 Codage des exposants

### 7.1.1 Présentation générale

L'information réelle véhiculée par le flux binaire AC-3 est constituée de coefficients de fréquence quantifiés. Ces coefficients sont remis au format virgule flottante, chaque coefficient comportant un exposant et une mantisse. Le présent paragraphe décrit comment les exposants sont codés et empaquetés dans le flux binaire.

Les exposants sont des valeurs de 5 bits indiquant le nombre de zéros de tête dans la représentation binaire d'un coefficient de fréquence. L'exposant constitue pour la mantisse un facteur d'échelle de valeur  $2^{-\text{exp}}$ . Les valeurs d'exposants autorisées sont comprises entre 0 (pour les coefficients de valeur la plus élevée ne comportant pas de zéros en tête) et 24 inclus. Pour les coefficients comportant plus de 24 zéros en tête, les exposants sont bloqués à 24 et les mantisses correspondantes sont autorisées à comporter des zéros en tête. Les exposants nécessitent 5 bits afin de représenter toutes les valeurs autorisées.

Les flux binaires AC-3 contiennent des exposants codés pour tous les canaux indépendants et tous les canaux couplés, ainsi que pour le canal de couplage et le canal d'effets basse fréquence lorsque ces derniers sont activés. Comme il n'existe pas d'information audio commune entre trames, le bloc 0 de chaque trame comporte de nouveaux exposants pour chaque canal. L'information d'exposant peut être commune aux blocs à l'intérieur d'une même trame, de sorte que les blocs 1 à 5 peuvent réutiliser les exposants de blocs antérieurs.

La transmission des exposants AC-3 fait appel à un codage différentiel dans lequel les exposants d'un canal sont codés d'une manière différentielle pour les diverses fréquences. Le premier exposant d'un canal de largeur de bande complète ou d'effets lfe est toujours envoyé sous la forme d'une valeur absolue de 4 bits, offrant une plage de valeurs entières de 0 à 15. La valeur indique le nombre de zéros de tête du premier coefficient de la transformée (terme continu). Les exposants suivants (par fréquences croissantes) sont transmis comme valeurs différentielles qui doivent être additionnées à l'exposant précédent pour former la valeur absolue suivante.

Les exposants différentiels sont combinés en groupes dans le bloc audio. Le groupement peut être fait selon l'une des trois méthodes D15, D25 ou D45 appelées stratégies d'exposant. Le nombre des exposants différentiels ainsi groupés, placés dans le bloc audio pour un canal donné est fonction de la stratégie d'exposant et de l'information de largeur de bande fréquences pour ce canal. Le nombre d'exposants dans chaque groupe ne dépend que de la stratégie d'exposant choisie.

Un bloc audio AC-3 possède deux types de champs contenant une information d'exposant. Le premier type définit la stratégie de codage d'exposant pour chaque canal alors que le second type contient les exposants réels pour les canaux nécessitant de nouveaux exposants. Pour les canaux indépendants, l'information de largeur de bande fréquences est fournie en même temps que les champs de stratégie d'exposant. Pour les canaux couplés et le canal de couplage, l'information de fréquence se trouve dans les champs décrivant la stratégie de couplage.

### 7.1.2 Stratégie d'exposant

L'information de stratégie d'exposant de chaque canal est incluse dans chaque bloc audio AC-3. L'information n'est jamais partagée entre trames, de sorte que le bloc 0 contient toujours une indication de stratégie (D15, D25 ou D45) pour chaque canal. Les blocs 1 à 5 peuvent indiquer la réutilisation d'exposants antérieurs dans la même trame. Les trois stratégies d'exposant sont un compromis entre le taux binaire nécessaire pour la transmission des exposants et leur résolution en fréquence. Le mode D15 offre la résolution en fréquence la plus fine, alors que le mode D45 nécessite le moins de données. Dans les trois modes, un certain nombre d'exposants sont combinés en mots de 7 bits lorsqu'ils sont codés dans un bloc audio. Le nombre d'exposants différentiels combinés entre eux constitue la principale différence entre les modes.

Les exposants absolus, présents dans le flux binaire au début des ensembles d'exposants en codage différentiel, sont envoyés sous la forme de valeurs de 4 bits et sont limités, soit en gamme de valeur, soit en résolution, afin d'économiser un bit. Pour les canaux de largeur de bande complète et d'effets lfe, l'exposant absolu initial de 4 bits représente une valeur de 0 à 15. Les valeurs d'exposants supérieures sont bloquées à 15. Pour le canal couplé, l'exposant absolu de 5 bits est limité aux valeurs paires et le bit le moins significatif n'est pas transmis. La résolution a été limitée à 0, 2, 4 ... 24. Chaque exposant différentiel peut prendre une des cinq valeurs  $-2$ ,  $-1$ ,  $0$ ,  $+1$ ,  $+2$ . Ceci permet des deltas atteignant au maximum  $\pm 2$  ( $\pm 12$  dB) entre exposants. Ces cinq valeurs sont mises en correspondance avec les valeurs 0, 1, 2, 3, 4 avant leur regroupement, comme le montre le Tableau 15.

TABLEAU 15

**Table de correspondance des valeurs d'exposants différentiels, mode D15**

Exposant différentiel	Valeur correspondante
+2	4
+1	3
0	2
-1	1
-2	0

Valeur correspondante: exposant différentiel + 2

Exposant différentiel: valeur correspondante - 2

Dans le mode D15, la mise en correspondance ci-dessus est appliquée pour le codage dans le flux binaire à chaque exposant différentiel isolé. Dans le mode D25, chaque paire d'exposants différentiels est représentée par une valeur mise en correspondance unique dans le flux binaire. Dans ce mode, le second exposant différentiel de chaque paire est donné par l'application d'une valeur delta de 0 sur le premier élément de la paire, comme le montre le Tableau 16.

TABLEAU 16

**Table de correspondance des valeurs d'exposants différentiels, mode D25**

Exposant différentiel $n$	Exposant différentiel $n + 1$	Valeur correspondante
+2	0	4
+1	0	3
0	0	2
-1	0	1
-2	0	0

Le mode D45 est comparable au mode D25, à ceci près que des ensembles de quatre exposants différentiels sont représentés par une valeur correspondante unique, comme le montre le Tableau 17.

TABLEAU 17

Table de correspondance des valeurs d'exposants différentiels, mode D45

Exposant différentiel $n$	Exposant différentiel $n + 1$	Exposant différentiel $n + 2$	Exposant différentiel $n + 3$	Valeur correspondante
+2	0	0	0	4
+1	0	0	0	3
0	0	0	0	2
-1	0	0	0	1
-2	0	0	0	0

Comme un exposant unique est en réalité partagé par 2 ou 4 mantisses différentes, les codeurs doivent garantir que l'exposant choisi pour la paire ou le quadruplet, a la valeur absolue minimale (correspondant à l'exposant le plus grand) nécessaire pour la représentation de toutes les mantisses.

Pour tous les modes, des ensembles de trois valeurs correspondantes (M1, M2 et M3) adjacentes (en fréquence) sont regroupées et codées pour donner une valeur de 7 bits selon la formule:

$$\text{Valeur du groupe codée à 7 bits} = (25 \times M1) + (5 \times M2) + M3$$

Le champ exposant d'un canal donné d'un bloc audio AC-3 est constitué d'un exposant absolu unique suivi d'un certain nombre de ces valeurs groupées.

### 7.1.3 Décodage des exposants

La stratégie d'exposant pour tout canal couplé ou indépendant est contenue dans un ensemble de champs de 2 bits dénommés `chexpstr[ch]`. Lorsque le canal de couplage est présent, un code de stratégie `cplexpstr` est également présent. Le Tableau 18 donne la correspondance entre le code de stratégie et la stratégie d'exposant.

TABLEAU 18

Stratégie de codage d'exposant

<code>chexpstr[ch]</code> , <code>cplexpstr</code>	Stratégie d'exposant	Nombre d'exposants par groupe
00	Réutiliser les exposants antérieurs	0
01	D15	3
10	D25	6
11	D45	12

Lorsque le canal d'effets basse fréquence est activé, le champ de stratégie `lfeexpstr` est présent. Il est décodé comme indiqué dans le Tableau 19.

TABLEAU 19

Stratégie de codage d'exposant lfe

<code>lfeexpstr</code>	Stratégie d'exposant	Nombre d'exposants par groupe
0	Réutiliser les exposants antérieurs	0
1	D15	3

Un ensemble de codes de largeur de bande de canal  $chbwcod[ch]$  suit les champs de stratégie d'exposant dans le flux binaire. Ces codes ne sont présents que pour les canaux indépendants (canaux non couplés) qui ont de nouveaux exposants dans le bloc en cours. Le code de largeur de bande canal définit le rang de la dernière mantisse de ce canal selon la formule:

$$endmant[ch] = ((chbwcod[ch] + 12) * 3) + 37; \quad /* \text{(le canal n'est pas couplé)} */$$

Pour les canaux couplés, le rang de la dernière mantisse est défini par le rang de départ du canal de couplage:

$$endmant[ch] = cplstrtmant; \quad /* \text{(le canal est couplé)} */$$

où  $cplstrtmant$  est calculé comme ci-dessous. Par définition, le rang de la première mantisse des canaux indépendants et couplés est égal à 0:

$$strtmant[ch] = 0;$$

Pour le canal de couplage, l'information de largeur de bande en fréquence est calculée à partir des champs  $cplbegf$  et  $cplendf$  appartenant à l'information de stratégie de couplage. Les rangs de départ et de fin des mantisses du canal de couplage sont définis par:

$$\begin{aligned} cplstrtmant &= (cplbegf * 12) + 37; \\ cplendmant &= ((cplendf + 3) * 12) + 37; \end{aligned}$$

Le canal d'effets basse fréquence, s'il est présent, commence toujours au rang 0 et a toujours le même nombre de mantisses:

$$\begin{aligned} lfestrtmant &= 0; \\ lfeendmant &= 7; \end{aligned}$$

Le deuxième ensemble de champs contient les exposants codés pour tous les canaux signalés comme ayant de nouveaux exposants dans le bloc en cours. Ces champs sont désignés par  $exps[ch][grp]$  pour les canaux indépendants et couplés, par  $cplexps[grp]$  pour le canal de couplage et par  $lfeexps[grp]$  pour le canal d'effets basse fréquence. Le premier élément des champs  $exps$  ( $exps[ch][0]$ ) et le champ  $lfeexps$  ( $lfeexps[0]$ ) est toujours un nombre absolu de 4 bits. Pour ces canaux, l'exposant absolu contient la valeur de l'exposant du premier coefficient (bin  $\neq 0$ ) de la transformée. Ces valeurs de 4 bits correspondent à des exposants de 5 bits dont la plage de valeurs a été limitée (de 0 à 15, au lieu de 0 à 24), c'est-à-dire que le bit le plus significatif est nul. L'exposant absolu du canal couplé  $cplabsexp$  est utilisé uniquement comme référence de départ du décodage des exposants différentiels du canal de couplage (c'est-à-dire qu'il ne représente pas réellement un exposant). L'exposant  $cplabsexp$  est présent dans le bloc avec une valeur de 4 bits mais correspond à une valeur de 5 bits. Le bit le moins significatif de l'exposant initial du canal couplé est toujours nul, de sorte que le décodeur doit prendre la valeur de 4 bits et la multiplier par 2 (décalage d'un rang vers la gauche) de manière à obtenir la valeur d'origine à 5 bits.

Pour chaque ensemble d'exposants codés, le nombre d'exposants regroupés, en excluant le premier exposant absolu, est calculé de la manière suivante:

Pour des canaux indépendants et couplés:

$$\begin{aligned} nchgrps[grp] &= truncate ((endmant[grp] - 1) / 3); & /* \text{pour le mode D15} */ \\ &= truncate ((endmant[grp] - 1 + 3) / 6); & /* \text{pour le mode D25} */ \\ &= truncate ((endmant[grp] - 1 + 9) / 12); & /* \text{pour le mode D45} */ \end{aligned}$$

Pour le canal de couplage:

$$\begin{aligned} ncplgrps &= (cplendmant - cplstrtmant) / 3; & /* \text{pour le mode D15} */ \\ &= (cplendmant - cplstrtmant) / 6; & /* \text{pour le mode D25} */ \\ &= (cplendmant - cplstrtmant) / 12; & /* \text{pour le mode D45} */ \end{aligned}$$

Pour le canal d'effets basse fréquence:

$$nlfegrps = 2;$$

Le décodage d'un ensemble d'exposants codés groupés produit un ensemble d'exposants absolus de 5 bits. Les exposants sont décodés comme suit:

1. Chaque groupe de 7 bits de valeurs de mises en correspondance (**gexp**) est décodé en utilisant la procédure inverse du codage:
 
$$M1 = \text{truncate}(\text{gexp} / 25);$$

$$M2 = \text{truncate}((\text{gexp} \% 25) / 5);$$

$$M3 = (\text{gexp} \% 25) \% 5;$$
2. Chaque valeur de mise en correspondance est convertie en un exposant différentiel (**dexp**) par soustraction du déplacement de la mise en correspondance:
 
$$\text{dexp} = M - 2;$$
3. L'ensemble d'exposants différentiels est converti en exposants absolus par addition de chaque exposant différentiel avec l'exposant absolu de rang précédent:
 
$$\text{exp}[n] = \text{exp}[n-1] + \text{dexp}[n];$$
4. Pour les modes D25 et D45, chaque exposant absolu est copié vers les autres membres de la paire ou du quadruplet.

La procédure ci-dessus peut être résumée comme suit:

Pseudo-code
<pre> /* dépaqueter les valeurs mises en correspondance */ for (grp = 0; grp &lt; ngrps; grp++) {     expacc = gexp[grp];     dexp[grp * 3] = truncate (expacc / 25);     expacc = expacc - ( 25 * dexp[grp * 3]);     dexp[(grp * 3) + 1] = truncate (expacc / 5);     expacc = expacc - (5 * dexp[(grp * 3) + 1]);     dexp[(grp * 3) + 2] = expacc; }  /* valeurs mises en correspondance non biaisées */ for (grp = 0; grp &lt; (ngrps * 3); grp++) {     dexp[grp] = dexp[grp] - 2; }  /* conversion de différentiel en absolu */ prevexp = absexp; for (i = 0; i &lt; (ngrps * 3); i++) {     aexp[i] = prevexp + dexp[i];     prevexp = aexp[i]; }  /* expansion en tableau complet d'exposants absolus en utilisant la taille de groupe grpsize */ exp[0] = absexp; for (i = 0; i &lt; (ngrps * 3); i++) {     for (j = 0; j &lt; grpsize; j++)     {         exp[(i * grpsize) + j + 1] = aexp[i];     } } </pre>



où:

**ngrps:** nombre d'exposants groupés ( $nchgrps[ch]$ ,  $ncplgrps$ , ou  $nlfegrps$ )

**grpsize** = 1 pour D15

= 2 pour D25

= 4 pour D45

**absexp:** exposant absolu ( $exps[ch][0]$ ,  $(cplabsexp << 1)$ , ou  $lfeexps[0]$ )

Pour le canal de couplage, la matrice de résultat  $exp[n]$  doit subir un déplacement de manière à correspondre au bin initial de la mantisse de couplage:

$$cplexp[n + cplstrtmant] = exp[n + 1];$$

Pour les autres canaux,  $exp[n]$  correspond directement au tableau des exposants absolus de ce canal.

## 7.2 Affectation de bits

### 7.2.1 Présentation générale

Le sous-programme d'affectation de bits analyse, compte tenu des phénomènes de masquage, l'enveloppe spectrale du signal audio à coder en vue de déterminer le nombre de bits à assigner à chaque mantisse de coefficient de la transformée. Le codeur procède à une affectation globale de bits à partir d'un pool commun de bits, en considérant l'ensemble des canaux comme un tout. Il n'existe pas de préassignation de bits d'exposant ou de mantisse, ce qui permet au sous-programme de faire, compte tenu des exigences du signal, une affectation flexible de bits couvrant l'ensemble des canaux, des fréquences et des blocs audio.

L'affectation de bits utilise un modèle paramétrique de l'ouïe humaine pour évaluer en fonction de la fréquence un seuil de bruit permettant de distinguer les composantes spectrales audibles de celles qui ne le sont pas. Divers paramètres du modèle d'ouïe peuvent être ajustés par le décodeur en fonction des caractéristiques du signal. Par exemple, une courbe de masquage prototype est définie sous la forme deux lignes continues constituée par des segments successifs, chaque segment étant caractérisé par une pente et une ordonnée à l'origine. Le codeur choisit, pour chaque segment, une pente et une ordonnée à l'origine parmi plusieurs valeurs disponibles. Le codeur peut traiter par itérations un ou plusieurs de ces paramètres pour optimiser le résultat. L'affectation définitive de bits est calculée lorsque tous les seuils de niveau de bruit ont été choisis par le codeur. Les paramètres du modèle sont transmis au décodeur avec d'autres informations annexes. Le décodeur exécute le sous-programme en un seul passage.

Le seuil de niveau de bruit estimé est calculé pour 50 bandes de largeur non uniforme égale à 1/6 d'octave. La structure de bande, indépendante de la fréquence d'échantillonnage, est définie dans les tableaux du paragraphe ci-après. L'affectation de bits nécessaire pour chaque mantisse est déterminée par consultation de table basée sur la différence entre la densité spectrale de puissance du signal d'entrée, évaluée selon une échelle de fréquences uniforme à résolution fine, et le seuil de niveau de bruit estimé selon une échelle de fréquences de granularité grossière (correspondant aux bandes). En conséquence, le résultat de l'affectation de bits pour une mantisse particulière possède une granularité spectrale qui est fonction de la stratégie d'exposant utilisée. Plus précisément, une affectation de bits différente est calculée pour chaque mantisse avec un ensemble d'exposants D15, pour chaque paire de mantisses avec un ensemble d'exposants D25 et pour chaque quadruplet de mantisses avec un ensemble d'exposants D45.

Le décodeur doit calculer l'affectation de bits chaque fois que la stratégie d'exposant ( $chexpstr$ ,  $cplexpstr$ ,  $lfeexpstr$ ) pour un ou plusieurs canaux n'indique pas de réutilisation, ou chaque fois que les indicateurs  $baie$ ,  $srnoffste$  ou  $deltbaie$  sont égaux à 1. L'affectation de bits doit donc être mise à jour entre une fois par bloc et une fois tous les 6 blocs audio, toutes les valeurs intermédiaires étant possibles. Un ensemble complet de nouvelles informations d'affectation de bits est toujours transmis dans le bloc audio 0.

Etant donné que le programme d'affectation de bits doit produire des résultats identiques pour toutes réalisations de codeurs et de décodeurs, les opérations de recherche de table et de calcul en virgule fixe de chaque étape doivent être définies d'une manière exacte. Dans toute la présentation qui suit, il est fait usage d'une arithmétique en complément à 2 avec signe. Toutes les additions sont effectuées dans un accumulateur de 14 bits ou plus. Tous les résultats intermédiaires et valeurs mémorisées sont des valeurs de 8 bits.

## 7.2.2 Affectation de bits paramétrique

Le présent paragraphe décrit les sept étapes du programme de calcul du résultat de l'affectation paramétrique de bits dans le décodeur. L'approche esquissée ici commence avec un ensemble unique d'exposants couplés ou non couplés et traite pour chaque étape la totalité des données d'entrée avant de passer à l'étape suivante. Cette technique, appelée exécution verticale est conceptuellement simple à décrire et à réaliser. D'une autre manière, les sept étapes peuvent être exécutées horizontalement, auquel cas, des passages multiples à travers l'ensemble des sept étapes sont faites pour des sous-ensembles disjoints de l'ensemble des exposants en entrée.

Le choix entre le mode vertical et le mode horizontal dépend de l'importance relative du temps d'exécution par rapport à l'utilisation de la mémoire dans la réalisation finale. L'exécution verticale de l'algorithme est plus rapide du fait de la réduction du nombre de boucles et des en-têtes de sauvegarde du contexte. Par contre l'exécution horizontale nécessite moins de mémoire pour stocker les tableaux temporaires générés par chaque étape. Des types de mise en œuvre mixtes verticaux/horizontaux combinant les avantages de chaque technique sont également possibles.

### 7.2.2.1 Initialisation

Calcul des fréquences de début et de fin du canal à décoder. Ces informations sont calculées comme suit à partir de paramètres du flux binaire:

Pseudo-code	
<pre> /* pour les canaux de largeur fbw */ for(ch=0; ch&lt;nfchans; ch++) {     strtmant[ch] = 0;     if(chincpl[ch]) endmant[ch] = 37 + (12 × cplbegf); /* canal couplé */     else endmant[ch] = 37 + (3 × (chbwcod + 12)); /* canal non couplé */ }  /* pour le canal de couplage */ cplstrtmant = 37 + (12 × cplbegf); cplendmant = 37 + (12 × (cplendf + 3));  /* pour le canal d'effets lfe */ lfestartmant = 0; lfeendmant = 7; </pre>	

Etape de traitement de cas particulier:

Il faut évaluer tous les paramètres de déplacement du rapport S/N à partir du flux binaire avant de poursuivre la procédure d'initialisation. Il s'agit des déplacements  $csnroffst$ ,  $fsnroffst[ch]$  et  $lfefsnroffst$ . S'ils sont tous égaux à 0, alors tous les éléments du tableau de pointeurs d'affectation de bits  $bap[]$  doivent être mis à 0 et il n'est pas nécessaire de procéder à un autre traitement d'affectation de bits pour le bloc audio en cours de traitement.

Exécuter, comme suit, des consultations de table pour déterminer les valeurs des paramètres  $sdecay$ ,  $fdecay$ ,  $sgain$ ,  $dbknee$  et  $floor$  à partir de paramètres du flux binaire:

Pseudo-code	
$sdecay = slowdec[sdcycod];$	/* Tableau 20 */
$fdecay = fastdec[fdccod];$	/* Tableau 21 */
$sgain = slowgain[sgaincod];$	/* Tableau 22 */
$dbknee = dbpbtab[dbpbcod];$	/* Tableau 23 */
$floor = floortab[floorcod];$	/* Tableau 24 */

Pour les portions non couplées de canal de largeur de bande complète, procéder à l'initialisation suivante:

Pseudo-code
<pre> start = strtmant[ch]; end = endmant[ch]; lowcomp = 0; fgain = fastgain[fgaincod[ch]]; /* Tableau 25 */ snroffset[ch] = ((csnroffset - 15) &lt;&lt; 4 + fsnroffset[ch]) &lt;&lt; 2; </pre>

Pour le canal de couplage, procéder à l'initialisation suivante:

Pseudo-code
<pre> start = cplstrtmant; end = cplendmant; fgain = fastgain[cplfgaincod]; /* Tableau 25 */ snroffset = ((csnroffset - 15) &lt;&lt; 4 + cplfsnroffset) &lt;&lt; 2; if (cplleake) {     fastleak = (cplleak &lt;&lt; 8) + 768;     slowleak = (cplsleak &lt;&lt; 8) + 768; } </pre>

Pour le canal d'effets lfe, procéder à l'initialisation suivante:

Pseudo-code
<pre> start = lfestrtmant; end = lfeendmant; lowcomp = 0; fgain = fastgain[lfefgaincod]; snroffset = ((csnroffset - 15) &lt;&lt; 4 + lfefsnroffset) &lt;&lt; 2; </pre>

### 7.2.2.2 Mise en correspondance des exposants en densité psd

Cette étape met en correspondance les exposants décodés vers une fonction de 13 bits avec signe donnant la densité de puissance spectrale logarithmique (psd).

Pseudo-code
<pre> for (bin=start; bin&lt;end; bin++) {     psd[bin] = (3072 - (exp[bin] &lt;&lt; 7)); } </pre>

Comme  $\exp[k]$  peut prendre des valeurs entières comprises entre 0 et 24, la plage dynamique des valeurs de  $\text{psd}[]$  va de 0, pour le signal de niveau le plus faible, à 3072 pour le signal de niveau le plus élevé. La fonction résultante est représentée sur une échelle de fréquences linéaires avec une granularité fine.

### 7.2.2.3 Intégration de la psd

Cette étape de l'algorithme intègre les valeurs psd à granularité fine à l'intérieur de chacune d'un grand nombre de bandes de 1/6 d'octave. Le Tableau 26 donne les 50 valeurs de matrice pour `bndtab[]` et `bndsz[]`. La matrice `bndtab[]` donne le numéro de la première mantisse dans chaque bande. La matrice `bndsz[]` contient la largeur de chaque bande en nombre de mantisses incluses. Le Tableau 27 contient les 256 valeurs de matrice pour `masktab[]` donnant la correspondance des numéros de mantisse avec les numéros des bandes de 1/6 d'octave associées. Ces deux matrices contiennent des valeurs dupliquées qui peuvent ne pas être entièrement présentes dans une réalisation effective. Ces valeurs sont données ici uniquement dans le but de simplifier la présentation.

L'intégration des valeurs psd dans chacune des bandes est faite par addition logarithmique. L'addition logarithmique est réalisée en calculant la différence entre les deux opérandes et en utilisant la différence absolue divisée par 2 comme adresse pour la consultation d'une matrice de longueur 256, `latab[]` dont le contenu est donné dans le Tableau 28.

Pseudo-code
<pre> j = start; k = masktab[start]; do {     bndpsd[k] = psd[j];     j++;     for (i = j; i &lt; min(bndtab[k+1], end); i++)     {         bndpsd[k] = logadd(bndpsd[k], psd[j]);         j++;     }     k++; } while (end &gt; bndtab[k++]); logadd(a, b) {     c = a - b;     address = min((abs(c) &gt;&gt; 1), 255);     if (c &gt;= 0)     {         return(a + latab(address));     }     else     {         return(b + latab(address));     } } </pre>

### 7.2.2.4 Calcul de la fonction d'excitation

La fonction d'excitation est calculée en appliquant la courbe de masquage prototype choisie par le codeur (et transmise au décodeur) à la courbe de densité spectrale de puissance psd intégrée (`bndpsd[]`). Le résultat de ce calcul est ensuite réduit en amplitude par les paramètres de gain `fgain` et `sgain`, qui sont également extraits du flux binaire.

## Pseudo-code

```

bndstrt = masktab[start];
bndend = masktab[end - 1] + 1;
if (bndstrt == 0) /* pour les canaux fbw et lfe */
{ /* Note: ne pas exécuter calc_lowcomp() pour la dernière bande du canal lfe, (bin = 6) */
  lowcomp = calc_lowcomp(lowcomp, bndpsd[0], bndpsd[1], 0);
  excite[0] = bndpsd[0] - fgain - lowcomp;
  lowcomp = calc_lowcomp(lowcomp, bndpsd[1], bndpsd[2], 1);
  excite[1] = bndpsd[1] - fgain - lowcomp;
  begin = 7;
  for (bin = 2; bin < 7; bin++)
  {
    lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin);
    fastleak = bndpsd[bin] - fgain;
    slowleak = bndpsd[bin] - sgain;
    excite[bin] = (fastleak - lowcomp);
    if (bndpsd[bin] <= bndpsd[bin+1])
    {
      begin = bin + 1;
      break;
    }
  }
  for (bin = begin; bin < min(bndend, 22); bin++)
  {
    lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin);
    fastleak -= fdecay;
    fastleak = max(fastleak, bndpsd[bin] - fgain);
    slowleak -= sdecay;
    slowleak = max(slowleak, bndpsd[bin] - sgain);
    excite[bin] = max(fastleak - lowcomp, slowleak);
  }
  begin = 22;
}
else /* pour le canal de couplage */
{
  begin = bndstrt;
}
for (bin = begin; bin < bndend; bin++)
{
  fastleak -= fdecay;
  fastleak = max(fastleak, bndpsd[bin] - fgain);
  slowleak -= sdecay;
  slowleak = max(slowleak, bndpsd[bin] - sgain);
  excite[bin] = max(fastleak, slowleak);
}
calc_lowcomp(a, b0, b1, bin)
{
  if (bin < 7)
  {
    if ((b0 + 256) == b1);
    {
      a = 384;
    }
    else if (b0 > b1)
    {
      a = max(0, a - 64);
    }
  }
  else if (bin < 20)
  {
    if ((b0 + 256) == b1)
    {
      a = 320;
    }
    else if (b0 > b1)
    {
      a = max(0, a - 64);
    }
  }
  else
  {
    a = max(0, a - 128);
  }
  return(a);
}

```

### 7.2.2.5 Calcul de la courbe de masquage

Dans cette étape, on calcule la courbe de masquage (seuil de niveau de bruit) à partir de la fonction d'excitation comme indiqué ci-dessous. Le seuil d'audition  $hth[[]]$  est donné par le Tableau 29. Les variables  $fscod$  et  $dbpbcod$  sont reçues par le décodeur dans le flux binaire.

Pseudo-code
<pre> for (bin = bndstr; bin &lt; bndend; bin++) {     if (bndpsd[bin] &lt; dbknee)     {         excite[bin] += ((dbknee - bndpsd[bin]) &gt;&gt; 2);     }     mask[bin] = max(excite[bin], hth[fscod][bin]); } </pre>

### 7.2.2.6 Appliquer l'affectation de bits delta

L'information optionnelle d'affectation de bits delta,  $dba$ , dans le flux binaire permet au codeur de transmettre au décodeur une information annexe qui modifie directement la courbe de masquage obtenue par le sous-programme paramétrique. L'affectation de bits delta peut être activée par le codeur pour des blocs audio qui bénéficient d'une amélioration de la qualité audio lorsque l'affectation de bits par défaut est modifiée de façon adéquate. L'option d'affectation de bits delta est disponible pour tout canal de largeur de bande complète et le canal de couplage.

Dans le cas où l'affectation de bits delta n'est pas utilisée et où aucune information  $dba$  n'est incluse dans le flux binaire, le décodeur ne doit pas modifier l'affectation par défaut. Une manière de garantir la non-modification consiste à initialiser à 0 les variables d'affectation de bits  $cpldeltseg$  et  $deltseg[ch]$  au début de chaque trame. Cette méthode arrête immédiatement le traitement de l'affectation de bits delta (voir ci-dessous), à moins qu'une information  $dba$  (incluant  $cpldeltseg$  et  $deltseg[ch]$ ) ne soit présente dans le flux binaire.

L'information  $dba$  modifiant l'affectation de bits du décodeur est transmise comme information annexe. Les modifications d'affectation se présentent sous la forme d'ajustements faits, sur la courbe de masquage par défaut calculée par le décodeur. Les ajustements peuvent être faits par multiples de  $\pm 6$  dB. En moyenne, un ajustement de courbe de  $-6$  dB correspond à un accroissement de 1 bit de la résolution de toutes les mantisses de la bande de  $1/6$  d'octave considérée. Le code suivant indique comment la modification est effectuée pour un seul canal. Le calcul de modification est effectué sur le canal de couplage (dans ce cas,  $deltseg$  ci-dessus correspond à  $cpldeltseg$ ) et sur chaque canal de largeur de bande complète (auquel cas  $deltseg$  correspond à  $deltseg[ch]$ ).

Pseudo-code
<pre> if ((deltbae == 0)    (deltbae == 1)) {     band = 0;     for (seg = 0; seg &lt; deltseg+1; seg++)     {         band += deltoffst[seg];         if (deltba[seg] &gt;= 4)         {             delta = (deltba[seg] - 3) &lt;&lt; 7;         }         else         {             delta = (deltba[seg] - 4) &lt;&lt; 7;         }         for (k = 0; k &lt; deltlen[seg]; k++)         {             mask[band] += delta;             band++;         }     } } </pre>

### 7.2.2.7 Calcul de l'affectation de bits

Le calcul de la matrice de pointeurs d'affectation (`bap[]`) est exécuté dans cette étape. La courbe de masquage, ajustée de `snroffset` lors d'une étape précédente puis tronquée, est soustraite de la matrice `psd[]` définie avec une granularité fine. La différence est décalée de 5 positions vers la droite, puis utilisée comme adresse dans la matrice `baptab[]` de façon à obtenir l'affectation résultante. Le contenu de la matrice `baptab[]` est représenté dans le Tableau 30.

Le codeur oblige la somme des affectations de mantisses pour tous les canaux d'une trame à être inférieure ou égale au nombre total de bits de mantisse disponibles pour cette trame. Le codeur procède pour cela à des itérations sur les valeurs de déplacement `csnrofst` et `fsnrofst` (ou `cpifsnrofst` ou `lfefsnrofst` pour les canaux de couplage et d'effets basse fréquence) afin d'obtenir le résultat approprié. Le décodeur est assuré de recevoir une affectation de mantisse qui respecte les contraintes associées à un débit binaire de transmission fixe.

La matrice `bap[]` contient à la fin de cette étape une série de pointeurs de 4 bits. Les pointeurs indiquent le nombre de bits affectés à chaque mantisse. Le Tableau 31 donne la correspondance entre les valeurs de pointeurs `bap` et la précision de quantification.

```

Pseudo-code
i = start;
j = masktab[start];
do
{
  mask[j] -= snroffset;
  mask[j] -= floor;
  if (mask[j] < 0)
  {
    mask[j] = 0;
  }
  mask[j] &= 0x1fe0;
  mask[j] += floor;
  for (k = i; k < min(bndtab[j] + bndsz[j], end); k++)
  {
    address = (psd[i] - mask[j]) >> 5;
    address = min(63, max(0, address));
    bap[i] = baptab[address];
    i++;
  }
}
while (end > bndtab[j++]);

```

### 7.2.3 Tableaux d'affectation de bits

TABLEAU 20

Table des décroissances lentes, `slowdec[]`

Adresse	<code>slowdec[address]</code>
0	0x0f
1	0x11
2	0x13
3	0x15

TABLEAU 21

**Table des décroissances rapides, fastdec[]**

Adresse	fastdec[address]
0	0x3f
1	0x53
2	0x67
3	0x7b

TABLEAU 22

**Table des gains lents, slowgain[]**

Adresse	slowgain[address]
0	0x540
1	0x4d8
2	0x478
3	0x410

TABLEAU 23

**Table dB/bit, dbpbtabs[]**

Adresse	dbpbtabs[address]
0	0x000
1	0x700
2	0x900
3	0xb00

TABLEAU 24

**Table des seuils, floortabs[]**

Adresse	floortabs[address]
0	0x2f0
1	0x2b0
2	0x270
3	0x230
4	0x1f0
5	0x170
6	0x0f0
7	0xf800



TABLEAU 25

Table des gains rapides, **fastgain[]**

Adresse	fastgain[address]
0	0x080
1	0x100
2	0x180
3	0x200
4	0x280
5	0x300
6	0x380
7	0x400

TABLEAU 26

Tables de structure de bande, **bndtab[]**, **bndsz[]**

Numéro de bande	bndtab[band]	bndsz[band]	Numéro de bande	bndtab[band]	bndsz[band]
0	0	1	25	25	1
1	1	1	26	26	1
2	2	1	27	27	1
3	3	1	28	28	3
4	4	1	29	31	3
5	5	1	30	34	3
6	6	1	31	37	3
7	7	1	32	40	3
8	8	1	33	43	3
9	9	1	34	46	3
10	10	1	35	49	6
11	11	1	36	55	6
12	12	1	37	61	6
13	13	1	38	67	6
14	14	1	39	73	6
15	15	1	40	79	6
16	16	1	41	85	12
17	17	1	42	97	12
18	18	1	43	109	12
19	19	1	44	121	12
20	20	1	45	133	24
21	21	1	46	157	24
22	22	1	47	181	24
23	23	1	48	205	24
24	24	1	49	229	24

TABLEAU 27

Table de conversion du rang du numéro de bin vers le numéro de bande, **masktab[bin]**  
 $\text{bin} = (10 \times A) + B$

	B = 0	B = 1	B = 2	B = 3	B = 4	B = 5	B = 6	B = 7	B = 8	B = 9
A = 0	0	1	2	3	4	5	6	7	8	9
A = 1	10	11	12	13	14	15	16	17	18	19
A = 2	20	21	22	23	24	25	26	27	28	28
A = 3	28	29	29	29	30	30	30	31	31	31
A = 4	32	32	32	33	33	33	34	34	34	35
A = 5	35	35	35	35	35	36	36	36	36	36
A = 6	36	37	37	37	37	37	37	38	38	38
A = 7	38	38	38	39	39	39	39	39	39	40
A = 8	40	40	40	40	40	41	41	41	41	41
A = 9	41	41	41	41	41	41	41	42	42	42
A = 10	42	42	42	42	42	42	42	42	42	43
A = 11	43	43	43	43	43	43	43	43	43	43
A = 12	43	44	44	44	44	44	44	44	44	44
A = 13	44	44	44	45	45	45	45	45	45	45
A = 14	45	45	45	45	45	45	45	45	45	45
A = 15	45	45	45	45	45	45	45	46	46	46
A = 16	46	46	46	46	46	46	46	46	46	46
A = 17	46	46	46	46	46	46	46	46	46	46
A = 18	46	47	47	47	47	47	47	47	47	47
A = 19	47	47	47	47	47	47	47	47	47	47
A = 20	47	47	47	47	47	48	48	48	48	48
A = 21	48	48	48	48	48	48	48	48	48	48
A = 22	48	48	48	48	48	48	48	48	48	49
A = 23	49	49	49	49	49	49	49	49	49	49
A = 24	49	49	49	49	49	49	49	49	49	49
A = 25	49	49	49	0	0	0				



TABLEAU 29

Table des seuils d'audition, hth[fscod][band]

Numéro de bande	hth[0][band] ( $f_s = 48$ kHz)	hth[1][band] ( $f_s = 44,1$ kHz)	hth[2][band] ( $f_s = 32$ kHz)
0	0x04d0	0x04f0	0x0580
1	0x04d0	0x04f0	0x0580
2	0x0440	0x0460	0x04b0
3	0x0400	0x0410	0x0450
4	0x03e0	0x03e0	0x0420
5	0x03c0	0x03d0	0x03f0
6	0x03b0	0x03c0	0x03e0
7	0x03b0	0x03b0	0x03d0
8	0x03a0	0x03b0	0x03c0
9	0x03a0	0x03a0	0x03b0
10	0x03a0	0x03a0	0x03b0
11	0x03a0	0x03a0	0x03b0
12	0x03a0	0x03a0	0x03a0
13	0x0390	0x03a0	0x03a0
14	0x0390	0x0390	0x03a0
15	0x0390	0x0390	0x03a0
16	0x0380	0x0390	0x03a0
17	0x0380	0x0380	0x03a0
18	0x0370	0x0380	0x03a0
19	0x0370	0x0380	0x03a0
20	0x0360	0x0370	0x0390
21	0x0360	0x0370	0x0390
22	0x0350	0x0360	0x0390
23	0x0350	0x0360	0x0390
24	0x0340	0x0350	0x0380

Numéro de bande	hth[0][band] ( $f_s = 48$ kHz)	hth[1][band] ( $f_s = 44,1$ kHz)	hth[2][band] ( $f_s = 32$ kHz)
25	0x0340	0x0350	0x0380
26	0x0330	0x0340	0x0380
27	0x0320	0x0340	0x0370
28	0x0310	0x0320	0x0360
29	0x0300	0x0310	0x0350
30	0x02f0	0x0300	0x0340
31	0x02f0	0x02f0	0x0330
32	0x02f0	0x02f0	0x0320
33	0x02f0	0x02f0	0x0310
34	0x0300	0x02f0	0x0300
35	0x0310	0x0300	0x02f0
36	0x0340	0x0320	0x02f0
37	0x0390	0x0350	0x02f0
38	0x03e0	0x0390	0x0300
39	0x0420	0x03e0	0x0310
40	0x0460	0x0420	0x0330
41	0x0490	0x0450	0x0350
42	0x04a0	0x04a0	0x03c0
43	0x0460	0x0490	0x0410
44	0x0440	0x0460	0x0470
45	0x0440	0x0440	0x04a0
46	0x0520	0x0480	0x0460
47	0x0800	0x0630	0x0440
48	0x0840	0x0840	0x0450
49	0x0840	0x0840	0x04e0

TABLEAU 30

Table d'affectation de pointeurs de bits, **baptab[]**

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

TABLEAU 31

**Correspondance entre bap, niveaux de quantifieurs  
et nombre de bits de la mantisse**

bap	Niveaux de quantifieurs	Bits de mantisse (bits du groupe / nombre de mantisses par groupe)
0	0	0
1	3	1,67 (5/3)
2	5	2,33 (7/3)
3	7	3
4	11	3,5 (7/2)
5	15	4
6	32	5
7	64	6
8	128	7
9	256	8
10	512	9
11	1024	10
12	2048	11
13	4096	12
14	16384	14
15	65536	16

### 7.3 Quantification et décodage des mantisses

#### 7.3.1 Présentation générale

Toutes les mantisses sont quantifiées à un niveau de précision déterminé par le pointeur bap correspondant. Les mantisses quantifiées à 15 niveaux ou moins utilisent une quantification symétrique. Les mantisses quantifiées à plus de 15 niveaux utilisent une quantification asymétrique qui est une représentation habituelle en complément à deux.

Un certain nombre de valeurs de mantisses quantifiées sont regroupées et codées dans un mot de code commun. Dans le cas du quantifieur à 3 niveaux, 3 valeurs quantifiées sont regroupées et représentées dans le flux binaire par un mot de code de 5 bits. Dans le cas du quantifieur à 5 niveaux, 3 valeurs quantifiées sont regroupées et représentées par un mot de code de 7 bits. Pour le quantifieur à 11 niveaux, 2 valeurs sont regroupées et représentées par un mot de code de 7 bits.

Dans le codeur, chaque coefficient de la transformée (qui est toujours  $< 1,0$ ) est cadré à gauche en décalant vers la gauche sa représentation binaire du nombre de positions indiqué par son exposant (de 0 à 24 décalages à gauche). Le coefficient amplifié est ensuite quantifié à un nombre de niveaux indiqué par le pointeur bap correspondant.

Le Tableau 32 donne le quantifieur à utiliser pour chaque bap. Si un pointeur bap est égal à 0, aucun bit n'est envoyé à la mantisse. Le regroupement est utilisé pour les valeurs de bap 1, 2 et 4 (quantifieurs à 3, 5 et 11 niveaux).

TABLEAU 32

## Correspondance entre bap et quantifieur

bap	Niveaux de quantifieurs	Type de quantification	Bits de mantisse (qntztab[bap]) (bits du groupe / nombre de mantisses par groupe)
0	0	Aucun	0
1	3	Symétrique	1,67 (5/3)
2	5	Symétrique	2,33 (7/3)
3	7	Symétrique	3
4	11	Symétrique	3,5 (7/2)
5	15	Symétrique	4
6	32	Asymétrique	5
7	64	Asymétrique	6
8	128	Asymétrique	7
9	256	Asymétrique	8
10	512	Asymétrique	9
11	1024	Asymétrique	10
12	2048	Asymétrique	11
13	4096	Asymétrique	12
14	16384	Asymétrique	14
15	65536	Asymétrique	16

Au cours du processus de décodage, le flux de données de mantisses est décomposé en mantisses individuelles de longueur variable, dans lesquelles sont intercalés des groupes codés de deux ou trois mantisses. Les mantisses de chaque ensemble d'exposants se présentent dans le flux binaire par ordre de fréquences croissantes. Toutefois les groupes se présentent à la position de la première mantisse contenue dans le groupe. Aucune opération de dépaquetage n'est exécutée dans le flux binaire pour les mantisses subséquentes du groupe.

### 7.3.2 Développement des mantisses pour une quantification asymétrique ( $6 \leq \text{bap} \leq 15$ )

Pour des valeurs de la matrice des pointeurs d'affectation de bits,  $6 \leq \text{bap} \leq 15$ , on utilise une quantification fractionnaire asymétrique en complément à deux. Chaque mantisse, et son exposant, sont la représentation en virgule flottante d'un coefficient de transformée. La virgule décimale est considérée comme se trouvant à gauche du bit le plus significatif, ce qui fait que le mot de mantisse représente une plage de valeurs de:

$$(1,0 - 2^{-\text{qntztab}[\text{bap}]_1}) \text{ à } -1,0.$$

La mantisse de rang  $k$  et de longueur  $\text{qntztab}[\text{bap}[k]]$  est extraite du flux binaire. La conversion inverse vers une représentation en virgule fixe est réalisée en décalant la mantisse vers la droite d'un nombre de positions égal à son exposant. Ce processus est représenté par l'expression suivante:

$$\text{transform\_coefficient}[k] = \text{mantissa}[k] \gg \text{exponent}[k];$$

Aucun groupage n'est fait pour des mantisses quantifiées d'une manière asymétrique.

### 7.3.3 Développement de mantisses pour une quantification symétrique ( $1 \leq \text{bap} \leq 5$ )

Les mantisses sont représentées par des valeurs codées lorsque la valeur du pointeur bap est comprise entre 1 et 5 ( $1 \leq \text{bap} \leq 5$ ). Les valeurs codées sont converties en mots binaires fractionnaires sous une représentation classique en complément à deux par une consultation de table. Le nombre de bits indiqué par le pointeur bap de la mantisse est extrait

du flux binaire et cadré à droite. Cette valeur codée est utilisée comme index d'une table de valeurs de la mantisse. La mantisse obtenue est décalée vers la droite d'un nombre de positions égal à son exposant pour donner la valeur du coefficient de la transformée:

$$\text{transform\_coefficient}[k] = \text{quantization\_table}[\text{mantissa\_code}[k]] \gg \text{exponent}[k];$$

La correspondance entre la valeur codée de la mantisse et la valeur réelle de la mantisse est donnée par les Tableaux 33 à 37.

#### 7.3.4 Bruit de désambiguïsation (dither) pour les mantisses de 0 bit (bap = 0)

Le décodeur AC-3 utilise des valeurs de bruit aléatoire (bruit de désambiguïsation (dither)) et non pas des valeurs quantifiées lorsque le nombre de bits affectés à la mantisse est nul (bap = 0). L'utilisation de la valeur aléatoire est conditionnée par la valeur de l'indicateur dithflag. Lorsque la valeur de dithflag est 1, la valeur du bruit aléatoire est utilisée. Lorsque la valeur de dithflag est 0, une valeur vraie nulle est utilisée. La variable dithflag est présente pour chaque canal. Le bruit de désambiguïsation (dither) est appliqué après que les canaux individuels ont été extraits du canal de couplage, afin d'éviter une corrélation entre les bruits de désambiguïsation (dither) appliqués aux fréquences hautes de chaque canal.

Il est possible d'utiliser toute séquence aléatoire habituelle pour produire les valeurs du bruit de désambiguïsation (dither). La longueur de mot de la valeur de bruit de désambiguïsation (dither) n'est pas critique et huit bits suffisent. Le facteur d'échelle optimal pour les mots de bruit de désambiguïsation (dither) consiste à prendre une distribution uniforme entre  $-1$  et  $+1$  et à lui appliquer un facteur multiplicatif de 0,707 donnant ainsi une distribution uniforme entre  $-0,707$  et  $+0,707$ . Un multiplicateur de 0,75 est suffisamment proche pour être également considéré comme optimal. Un multiplicateur de 0,5 donnant une distribution uniforme entre  $-0,5$  et  $+0,5$  est également acceptable.

Après qu'une valeur de bruit de désambiguïsation (dither) a été assignée à la mantisse, celle-ci est décalée vers la droite d'un nombre de positions égal à son exposant pour donner la valeur du coefficient de la transformée:

$$\text{transform\_coefficient}[k] = \text{scaled\_dither\_value} \gg \text{exponent}[k];$$

TABLEAU 33

#### Quantification pour bap = 1 (3 niveaux)

Code de mantisse	Valeur de mantisse
0	$-2./3$
1	0
2	$2./3$

TABLEAU 34

#### Quantification pour bap = 2 (5 niveaux)

Code de mantisse	Valeur de mantisse
0	$-4./5$
1	$-2./5$
2	0
3	$2./5$
4	$4./5$



TABLEAU 35

**Quantification pour bap = 3 (7 niveaux)**

Code de mantisse	Valeur de mantisse
0	-6./7
1	-4./7
2	-2./7
3	0
4	2./7
5	4./7
6	6./7

TABLEAU 36

**Quantification pour bap = 4 (11 niveaux)**

Code de mantisse	Valeur de mantisse
0	-10./11
1	-8./11
2	-6./11
3	-4./11
4	-2./11
5	0
6	2./11
7	4./11
8	6./11
9	8./11
10	10./11

TABLEAU 37

**Quantification pour bap = 5 (15 niveaux)**

Code de mantisse	Valeur de mantisse
0	-14./15
1	-12./15
2	-10./15
3	-8./15
4	-6./15
5	-4./15
6	-2./15
7	0
8	2./15
9	4./15
10	6./15
11	8./15
12	10./15
13	12./15
14	14./15

### 7.3.5 Dégroupage de mantisses

Lorsque  $bap = 1, 2$  ou  $4$ , les valeurs codées des mantisses subissent une compression supplémentaire par combinaison des mots de niveau 3 et de niveau 5 en groupes distincts représentant trois mantisses et les mots de niveau 11 en groupes représentant deux mantisses. Les groupes sont remplis dans l'ordre de traitement des mantisses. Les groupes sont partagés entre les ensembles d'exposants si le nombre de mantisses d'un ensemble d'exposants ne remplit pas complètement un groupe. L'ensemble d'exposants suivant du bloc continue à remplir les groupes partiels. Le dernier groupe d'un bloc est rempli avec des mantisses fictives si le nombre total de coefficients de transformée obtenus par quantification des niveaux 3 ou 5 n'est pas divisible par 3 ou si le nombre de mots de niveau 11 n'est pas divisible par 2. Ces mantisses fictives sont ignorées par le décodeur. Les groupes sont extraits du flux binaire en utilisant la longueur obtenue à partir du pointeur  $bap$ . Les mantisses quantifiées de niveau 3 ( $bap = 1$ ) sont quantifiées en triplets de 5 bits. Les mantisses quantifiées de niveau 5 ( $bap = 2$ ) sont quantifiées en triplets de 7 bits. Les mantisses quantifiées de niveau 11 ( $bap = 1$ ) sont quantifiées en paires de 7 bits.

#### Equations de codage

$bap = 1$ :

$$\text{group\_code} = 9 * \text{mantissa\_code}[a] + 3 * \text{mantissa\_code}[b] + \text{mantissa\_code}[c];$$

$bap = 2$ :

$$\text{group\_code} = 25 * \text{mantissa\_code}[a] + 5 * \text{mantissa\_code}[b] + \text{mantissa\_code}[c];$$

$bap = 4$ :

$$\text{group\_code} = 11 * \text{mantissa\_code}[a] + \text{mantissa\_code}[b];$$

#### Equations de décodage

$bap = 1$ :

$$\begin{aligned} \text{mantissa\_code}[a] &= \text{truncate}(\text{group\_code} / 9); \\ \text{mantissa\_code}[b] &= \text{truncate}((\text{group\_code} \% 9) / 3); \\ \text{mantissa\_code}[c] &= (\text{group\_code} \% 9) \% 3; \end{aligned}$$

$bap = 2$ :

$$\begin{aligned} \text{mantissa\_code}[a] &= \text{truncate}(\text{group\_code} / 25); \\ \text{mantissa\_code}[b] &= \text{truncate}((\text{group\_code} \% 25) / 5); \\ \text{mantissa\_code}[c] &= (\text{group\_code} \% 25) \% 5; \end{aligned}$$

$bap = 4$ :

$$\begin{aligned} \text{mantissa\_code}[a] &= \text{truncate}(\text{group\_code} / 11); \\ \text{mantissa\_code}[b] &= \text{group\_code} \% 11; \end{aligned}$$

la mantisse a précède la mantisse b qui précède la mantisse c.

## 7.4 Couplage de canal

### 7.4.1 Présentation générale

Le couplage de canal, s'il est activé, est réalisé au moment du codage en faisant la moyenne des coefficients de la transformée pour l'ensemble des canaux inclus dans le canal de couplage. Chaque canal couplé possède un ensemble unique de coordonnées de couplage utilisées afin de préserver les enveloppes des fréquences hautes des canaux d'origine. Le couplage est réalisé au-dessus d'une fréquence de couplage définie par la valeur  $cplgegf$ .

Le décodeur reconvertit le canal de couplage en canaux individuels en multipliant les valeurs des coefficients de transformée du canal couplé par la coordonnée de couplage de ce canal et sous-bande fréquence. Une étape supplémentaire de traitement est présente dans le mode 2/0. Si le bit  $phsflginu$  est égal à 1 ou si l'état équivalent se propage à partir du bloc précédent, alors des bits de remise en phase sont envoyés dans le flux binaire au moyen des bits d'indicateur de phase. Les bits d'indicateur de phase représentent les sous-bandes de couplage par ordre de fréquences ascendantes. Si un bit d'indicateur de phase d'une sous-bande particulière est égal à 1, tous les coefficients de la transformée du canal droit de cette sous-bande sont multipliés par  $-1$  après modification par la coordonnée de couplage, mais avant la transformation inverse.

### 7.4.2 Structure de sous-bande pour le couplage

Les coefficients de la transformée de rang 37 à 252 sont groupés en 18 sous-bandes de 12 coefficients chacune comme indiqué dans le Tableau 38. Le paramètre `cplbegf` indique le numéro de la sous-bande couplage pour lequel commence le processus de couplage. Tous les canaux sont codés d'une manière indépendante en dessous de la fréquence (ou rang du coefficient de la transformée) indiquée par `cplbegf`. Au-dessus de la fréquence indiquée par `cplbegf`, les canaux inclus dans le processus de couplage (`chincpl[ch] = 1`) partagent le canal commun de couplage jusqu'à la fréquence (ou rang du coefficient de la transformée) indiquée par `cplendf` qui donne la dernière sous-bande couplage qui est codée. Le paramètre `cplendf` est interprété en additionnant 2 à sa valeur, de sorte que le rang de la dernière sous-bande codée peut prendre des valeurs de 2 à 17.

Les sous-bandes de couplage sont combinées en bandes de couplage pour lesquelles des coordonnées de couplage sont générées et incluses dans le flux binaire. La structure de la bande couplage est définie par `cplbndstrc[sbnd]`. Chaque bit du tableau `cplbndstrsc[]` indique si la sous-bande référencée est combinée dans la bande couplage précédente (de fréquence inférieure). Les bandes de couplage sont ainsi constituées d'un nombre entier de sous-bandes de couplage (voir le § 5.4.3.13).

TABLEAU 38  
Sous-bandes couplage

Rang de la sous-bande de couplage	Rang ct inférieur	Rang ct supérieur	Coupure BF (kHz) @ $f_s = 48$ kHz	Coupure HF (kHz) @ $f_s = 48$ kHz	Coupure BF (kHz) @ $f_s = 44,1$ kHz	Coupure HF (kHz) @ $f_s = 44,1$ kHz
0	37	48	3,42	4,55	3,14	4,18
1	49	60	4,55	5,67	4,18	5,21
2	61	72	5,67	6,80	5,21	6,24
3	73	84	6,80	7,92	6,24	7,28
4	85	96	7,92	9,05	7,28	8,31
5	97	108	9,05	10,17	8,31	9,35
6	109	120	10,17	11,30	9,35	10,38
7	121	132	11,30	12,42	10,38	11,41
8	133	144	12,42	13,55	11,41	12,45
9	145	156	13,55	14,67	12,45	13,48
10	157	168	14,67	15,80	13,48	14,51
11	169	180	15,80	16,92	14,51	15,55
12	181	192	16,92	18,05	15,55	16,58
13	193	204	18,05	19,17	16,58	17,61
14	205	216	19,17	20,30	17,61	18,65
15	217	228	20,30	21,42	18,65	19,68
16	229	240	21,42	22,55	19,68	20,71
17	241	252	22,55	23,67	20,71	21,75

ct: coefficient de transformée.

NOTE 1 – Pour une fréquence d'échantillonnage de 32 kHz, les fréquences limites des sous-bandes sont les 2/3 des valeurs données pour 48 kHz.

### 7.4.3 Format de coordonnée de couplage

Il existe des coordonnées de couplage pour chaque bande de couplage `[bnd]` dans chaque canal `[ch]` couplé (`chincp[ch]==1`). Les coordonnées de couplage sont transmises avec une virgule flottante. L'exposant est transmis sous la forme d'une valeur à 4 bits (`cplcoexp[ch][bnd]`) indiquant le nombre de décalage à droite qui doit être appliqué à la valeur fractionnaire de la mantisse. Les mantisses sont transmises sous la forme de valeurs à 4 bits

(`cplcomant[ch][bnd]`) qui doivent être correctement mises à l'échelle avant d'être utilisées. Les mantisses sont des valeurs non signées, d'où l'absence de bit de signe. Sauf dans le cas limite où la valeur de l'exposant est égale à 15, la valeur de la mantisse est toujours comprise entre 0,5 et 1,0. Donc, lorsque la valeur de l'exposant est inférieure à 15, le bit le plus significatif de la mantisse est toujours égal à «1» et n'est pas transmis; les 4 bits suivants de la mantisse sont transmis. Cela donne un bit de précision supplémentaire. Lorsque la valeur de l'exposant est égale à 15, celle de la mantisse est obtenue en divisant la valeur à 4 bits de `cplcomant` par 16. Lorsque la valeur de l'exposant est inférieure à 15, la valeur de la mantisse est calculée en ajoutant 16 à la valeur à 4 bits de `cplcomant` puis en divisant la somme par 32.

La plage dynamique des coordonnées de couplage est augmentée au-delà des possibilités offertes par l'exposant à 4 bits au moyen d'une coordonnée directrice de couplage à 2 bits par canal (`mstrcplco[ch]`) qui est utilisée pour placer toutes les coordonnées de couplage à l'intérieur de ce canal. Les valeurs des exposants pour chaque canal sont augmentées d'une valeur égale à 3 fois la valeur de `mstrcplco` applicable à ce canal. Cela augmente la plage dynamique des coordonnées de couplage de 54 dB supplémentaires.

Le pseudo-code suivant indique comment obtenir la coordonnée de couplage (`cplco`) pour chaque bande de couplage [`bnd`] dans chaque canal [`ch`].

Pseudo-code
<pre> if (cplcoexp[ch, bnd] == 15) {     cplco_temp[ch, bnd] = cplcomant[ch, bnd] / 16 ; } else {     cplco_temp[ch, bnd] = (cplcomant[ch, bnd] + 16) / 32 ; } cplco[ch, bnd] = cplco_temp[ch, bnd]&gt;&gt; (cplcoexp[ch, bnd] + 3 * mstrcplco[ch]) ; </pre>

Au moyen de la matrice `cplbndstrc[]`, les valeurs des coordonnées de couplage qui s'appliquent aux bandes de couplage sont converties (en reproduisant les valeurs indiquées par la valeur «1» dans `cplbandstrc[]`) en valeurs applicables aux sous-bandes de couplage.

Les mantisses des canaux individuels sont ensuite reconstituées à partir du canal couplé comme suit:

Pseudo-code
<pre> for (sbnd = cplbegf; sbnd &lt; 3 + cplendf; sbnd++) {     for (bin = 0; bin &lt; 12; bin++)     {         chmant[ch, sbnd*12+bin+37] = cplmant[sbnd*12+bin+37] * cplco[ch, sbnd] * 8;     } } </pre>

## 7.5 Rematriçage

### 7.5.1 Présentation générale

Le rematriçage du codage AC-3 est une technique de combinaison des canaux dans laquelle sont codées les sommes et les différences de canaux fortement corrélés plutôt que les canaux eux-mêmes. Ainsi, plutôt que de coder et d'empaqueter les canaux gauche et droit dans un codeur à deux canaux, on calcule:

$$\text{left}' = 0.5 * (\text{left} + \text{right});$$

$$\text{right}' = 0.5 * (\text{left} - \text{right});$$

Les opérations habituelles de quantification et d'empaquetage des données sont ensuite effectuées sur *left'* et *right'*. Il est clair que si les signaux stéréo initiaux étaient identiques sur les deux canaux (c'est-à-dire mono sur deux canaux), cette technique fournira un signal *left'* identique au signal gauche initial et un signal *right'* identiquement nul. Il devient donc possible de coder le canal *right'* avec très peu de bits et d'accroître la résolution du canal *left'*.

Cette technique est particulièrement importante pour préserver la compatibilité ambiophonique «Dolby Surround». Prenons par exemple une source mono à deux canaux telle que décrite ci-dessus. Un décodeur Dolby Pro Logic essayera de transférer vers le canal central toutes les informations en phase et vers le canal ambiophonique toutes les informations hors de phase. Si le rematriçage n'est pas activé, le décodeur Pro Logic reçoit les signaux suivants:

$$\text{Received left} = \text{left} + \text{QN1};$$

$$\text{Received right} = \text{right} + \text{QN2};$$

où QN1 et QN2 sont des séquences indépendantes (c'est-à-dire non corrélées) de bruits de quantification introduites par la quantification de l'algorithme de codage AC-3 et dépendent du programme. Le décodeur Pro Logic construit ensuite les canaux central et ambiophonique de la manière suivante:

$$\text{center} = 0.5 * (\text{left} + \text{QN1}) + 0.5 * (\text{right} + \text{QN2});$$

$$\text{surround} = 0.5 * (\text{left} + \text{QN1}) - 0.5 * (\text{right} + \text{QN2}); \quad /* \text{ en ignorant le déphasage de } 90 \text{ degrés } */$$

Pour le canal central, QN1 et QN2 s'additionnent mais restent masqués par le signal dominant droit + gauche. Pour le canal ambiophonique cependant, les canaux gauche et droit s'annulent et les haut-parleurs ambiophoniques reproduisent la différence des séquences de bruits de quantification (QN1 – QN2).

Si le rematriçage est actif, les canaux central et ambiophonique seront reproduits plus facilement:

$$\text{center} = \text{left}' + \text{QN1};$$

$$\text{surround} = \text{right}' + \text{QN2};$$

Dans ce cas, le bruit de quantification QN2 du canal ambiophonique est d'un niveau bien moindre et masqué par le signal *right'* de différence.

## 7.5.2 Définitions des bandes de fréquences

Le rematriçage dans le codage AC-3 est effectué d'une manière indépendante dans des bandes de fréquences distinctes. Il existe quatre bandes de fréquences dont les limites dépendent de l'information de couplage. Les limites sont définies par référence à un rang de coefficient de transformée et les limites correspondantes des bandes de fréquences pour le rematriçage changent avec la fréquence d'échantillonnage. Les Tableaux ci-dessous indiquent les bandes de fréquences de rematriçage pour des fréquences d'échantillonnage de 48 kHz et 44,1 kHz. Pour une fréquence d'échantillonnage de 32 kHz, les bandes de fréquences de rematriçage sont les 2/3 des valeurs indiquées pour 48 kHz.

### 7.5.2.1 Couplage non actif

Il existe 4 bandes de rematriçage ( $n_{\text{rematbd}} = 4$ ) si le couplage n'est pas actif ( $c_{\text{plinu}} = 0$ ).

TABLEAU 39

Bandes de rematriçage – Tableau A

Numéro de bande	Rang ct inférieur	Rang ct supérieur	Fréquence basse (kHz) $f_s = 48 \text{ kHz}$	Fréquence haute (kHz) $f_s = 48 \text{ kHz}$	Fréquence basse (kHz) $f_s = 44,1 \text{ kHz}$	Fréquence haute (kHz) $f_s = 44,1 \text{ kHz}$
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14
2	37	60	3,42	5,67	3,14	5,21
3	61	252	5,67	23,67	5,21	21,75

ct: coefficient de transformée

### 7.5.2.2 Couplage actif, $cplbegf > 2$

Si le couplage est actif ( $cplinu = 1$ ), et si  $cplbegf > 2$ , il existe 4 bandes de rematriçage ( $nrematbd = 4$ ). La quatrième et dernière bande de rematriçage se termine au point où commence le couplage.

TABLEAU 40

Bandes de rematriçage – Tableau B

Numéro de bande	Rang ct inférieur	Rang ct supérieur	Fréquence basse (kHz) $f_s = 48$ kHz	Fréquence haute (kHz) $f_s = 48$ kHz	Fréquence basse (kHz) $f_s = 44,1$ kHz	Fréquence haute (kHz) $f_s = 44,1$ kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14
2	37	60	3,42	5,67	3,14	5,21
3	61	A	5,67	B	5,21	C

ct: coefficient de transformée

$$A = 36 + cplbegf * 12$$

$$B = (A + 1/2) * 0,09375 \text{ kHz}$$

$$C = (A + 1/2) * 0,08613 \text{ kHz}$$

### 7.5.2.3 Couplage actif, $2 \geq cplbegf > 0$

Si le couplage est actif ( $cplinu = 1$ ), et si  $2 \geq cplbegf > 0$ , il existe 3 bandes de rematriçage ( $nrematbd = 3$ ). La troisième et dernière bande de rematriçage se termine au point où commence le couplage.

TABLEAU 41

Bandes de rematriçage – Tableau C

Numéro de bande	Rang ct inférieur	Rang ct supérieur	Fréquence basse (kHz) $f_s = 48$ kHz	Fréquence haute (kHz) $f_s = 48$ kHz	Fréquence basse (kHz) $f_s = 44,1$ kHz	Fréquence haute (kHz) $f_s = 44,1$ kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14
2	37	A	3,42	B	3,14	C

ct: coefficient de transformée

$$A = 36 + cplbegf * 12$$

$$B = (A + 1/2) * 0,09375 \text{ kHz}$$

$$C = (A + 1/2) * 0,08613 \text{ kHz}$$

### 7.5.2.4 Couplage actif, $cplbegf=0$

Si le couplage est actif ( $cplinu = 1$ ), et si  $cplbegf = 0$ , il existe 2 bandes de rematriçage ( $nrematbd = 2$ ).

TABLEAU 42

Bandes de rematriçage – Tableau D

Numéro de bande	Rang ct inférieur	Rang ct supérieur	Fréquence basse (kHz) $f_s = 48$ kHz	Fréquence haute (kHz) $f_s = 48$ kHz	Fréquence basse (kHz) $f_s = 44,1$ kHz	Fréquence haute (kHz) $f_s = 44,1$ kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14

ct: coefficient de transformée

### 7.5.3 Technique de codage

Le rematriçage est utilisé par le codeur si le mode 2/0 est sélectionné. Les carrés des coefficients sont nommés dans les bandes de fréquences de rematriçage définies précédemment pour les combinaisons L (canal gauche), R (canal droit), L + R et L - R.

Pseudo-code
<pre> if(minimum sum for a rematrixing sub-band n is L or R) {   the variable rematflg[n] = 0;   transmitted left = input L;   transmitted right = input R; } if(minimum sum for a rematrixing sub-band n is L+R or L-R) {   the variable rematflg[n] = 1;   transmitted left = 0.5* input (L+R);   transmitted right = 0.5* input (L-R); } </pre>

La sélection de la combinaison de matrices est faite bloc par bloc. La suite du processus de codage des canaux gauche et droite transmis est la même quel que soit l'état des indicateurs de rematriçage, 0 ou 1.

### 7.5.4 Technique de décodage

Un bit (indicateur de rematriçage), indiquant si les deux canaux ont subi le rematriçage dans la bande donnée, est envoyé dans le flux binaire pour chaque bande concernée. Si ce bit est nul, aucune opération supplémentaire n'est nécessaire. Si ce bit est positionné, le décodeur AC-3 effectue l'opération suivante pour rétablir les canaux individuels:

$$\begin{aligned} \text{left}(\text{band } n) &= \text{received left}(\text{band } n) + \text{received right}(\text{band } n); \\ \text{right}(\text{band } n) &= \text{received left}(\text{band } n) - \text{received right}(\text{band } n); \end{aligned}$$

Il est à noter que, si le couplage n'est pas actif, les deux canaux peuvent avoir des largeurs de bande différentes. Dans ce cas, le rematriçage n'est appliqué que jusqu'à la limite la plus basse de la largeur de bandes deux canaux. L'ensemble des quatre indicateurs de rematriçage est envoyé dans le flux binaire indépendamment de la largeur de bande réelle (dans l'hypothèse où le bit de stratégie de rematriçage est positionné).

## 7.6 Normalisation du dialogue

La syntaxe du flux AC-3 comporte des éléments qui permettent de satisfaire les auditeurs dans de nombreuses situations différentes. L'élément `dialnorm` permet une reproduction uniforme d'un dialogue lors du décodage de tout flux binaire AC-3.

### 7.6.1 Présentation générale

Quand des signaux audio issus de sources différentes sont reproduits, il arrive souvent que le niveau sonore apparent varie selon les sources. Les diverses sources peuvent être des segments distincts d'un programme (c'est-à-dire annonce commerciale intercalée pendant la diffusion d'un film), différents canaux de diffusion ou des supports différents (disque et bande). La technologie de codage AC-3 résout ce problème en codant explicitement une indication de niveau sonore dans le flux binaire AC-3.

La référence utilisée est le niveau subjectif d'un dialogue normal. Le mot de normalisation de dialogue de 5 bits, `dialnorm`, présent dans l'information BSI, indique le niveau sonore subjectif du dialogue normal par rapport au niveau numérique 100%. La valeur de 5 bits est interprétée comme un entier sans signe (le bit le plus significatif est transmis en tête) donnant une plage de valeurs possibles de 1 à 31. L'entier sans signe indique le plafond en dB au-dessus du niveau subjectif de dialogue. Cette valeur peut aussi être interprétée comme une indication du nombre de dB auquel le niveau subjectif de dialogue se situe au-dessous du niveau numérique 100%.

La valeur de `dialnorm` n'est pas utilisée directement par le décodeur AC-3, mais plutôt par la partie du système de reproduction sonore fixant le volume de la reproduction, c'est-à-dire la commande de volume du système. La commande de volume du système est en général réglée par l'auditeur qui fixe le volume souhaité ou le niveau de pression acoustique (SPL). L'auditeur règle une commande volume qui, en général, ajuste directement le gain du système de reproduction.

Avec un codage AC-3 et la valeur de *dialnorm*, le gain du système de reproduction devient une fonction à la fois du niveau de pression sonore souhaité par l'auditeur pour le dialogue et de la valeur de *dialnorm* indiquant le niveau de dialogue du signal audio. Il est ainsi possible pour l'auditeur de fixer d'une manière fiable le niveau du volume de dialogue et le niveau subjectif de dialogue restera le même quel que soit le programme AC-3 décodé.

*Exemple:*

L'auditeur règle la commande volume à 67 dB (il est possible avec la normalisation de dialogue AC-3 d'étalonner un réglage de volume du système directement en niveaux de pression sonore et cette indication conservera sa justesse pour toute source audio codée avec AC-3). Un programme de divertissement de haute qualité est reçu et le flux binaire AC-3 indique que le dialogue se situe à 25 dB au-dessous du niveau numérique 100%. Le système de reproduction fixe automatiquement le gain du système de reproduction de manière à ce qu'un signal numérique en pleine échelle soit reproduit avec une pression acoustique de 92 dB. Le dialogue (situé à -25 dB) sera ainsi reproduit avec un niveau de pression sonore de 67 dB.

Le programme de diffusion est interrompu par un message commercial de niveau sonore de -15 dB par rapport au niveau numérique 100%. Le gain du système baisse automatiquement, de telle façon que le niveau numérique 100% est reproduit à un niveau de pression sonore de 82 dB. Le dialogue du message commercial (situé à -15 dB) est reproduit, comme souhaité, à un niveau de pression acoustique de 67 dB.

Pour permettre au système de normalisation du dialogue de fonctionner, la valeur de *dialnorm* doit être communiquée par le décodeur AC-3 à la commande de gain du système de façon à ce que *dialnorm* puisse interagir avec le réglage de niveau sonore fait par l'auditeur. Si la fonction de commande de volume d'un système est faite par une multiplication numérique dans le décodeur AC-3, le volume choisi par l'utilisateur doit être communiqué au décodeur AC-3. Le réglage de volume choisi par l'auditeur et la valeur de *dialnorm* doivent être rapprochés et combinés de manière à ajuster le gain de sortie du système de reproduction.

Le réglage de la commande de volume du système n'est pas une fonction du codage AC-3. Le flux binaire AC-3 transporte l'information permettant au réglage de volume de système d'être réalisé d'une manière qui supprime automatiquement les différences de niveaux indésirables entre sources de programmes. Il est nécessaire que la valeur de *dialnorm* et le niveau de volume sonore sélectionné par l'utilisateur soient utilisés tous deux pour fixer le gain du système de reproduction.

## 7.7 Compression de plage dynamique

### 7.7.1 Commande de plage dynamique; *dynrng*, *dynrng2*

L'élément de commande de gain *dynrng* permet au diffuseur de programme de mettre en œuvre une réduction de plage dynamique subjectivement satisfaisante pour la majeure partie de l'audience tout en permettant à certains auditeurs de pouvoir utiliser une partie plus grande ou la totalité de la plage dynamique d'origine.

#### 7.7.1.1 Présentation générale

Un problème constant dans la diffusion de programmes audio est que certains auditeurs souhaitent des plages dynamiques différentes. Un programme d'origine de haute qualité (tels que des films de long métrage) est normalement mixé avec une plage dynamique très étendue. Si l'on utilise les dialogues comme référence, des sons forts tels qu'une explosion sont souvent à un niveau supérieur de 20 dB ou plus, et des sons faibles tels que le bruissement des feuilles peuvent être à un niveau inférieur de 50 dB. Il est désagréable dans beaucoup de situations d'écoute de permettre au son de devenir très bruyant, ce qui nécessite une compression vers le bas des niveaux des sons les plus bruyants. De même, les sons très faibles sont inaudibles dans beaucoup de situations d'écoute et leur niveau doit être relevé pour les rendre audibles. Comme la majorité des auditeurs se contente d'une plage dynamique réduite, les pistes sonores qui ont été mixées avec une plage dynamique étendue sont en général comprimées: la plage dynamique est réduite en abaissant le niveau des sons forts et en relevant le niveau des sons faibles. Si cela satisfait aux besoins de la majorité des auditeurs, cela supprime la possibilité, pour une partie des auditeurs de profiter du programme sonore d'origine dans la forme souhaitée. La technologie de codage audio AC-3 résout ce problème en permettant d'insérer dans le flux binaire AC-3 des valeurs de commande de la plage dynamique.

Les valeurs de commande de la plage dynamique, *dynrng*, indiquent une modification du gain devant être mise en œuvre dans le décodeur en vue de réaliser la compression de plage dynamique. Chaque valeur de *dynrng* peut indiquer une modification du gain de  $\pm 24$  dB. La séquence de valeurs de *dynrng* constitue un signal de commande de compression. Un codeur ou un équipement de traitement de flux binaire AC-3 génère la séquence de valeurs de *dynrng*. Chacune de ces valeurs est utilisée par le décodeur AC-3 pour modifier le gain d'un ou de plusieurs blocs audio. Les valeurs de *dynrng* indiquent normalement une réduction de gain pendant les passages les plus forts et un accroissement de gain pendant les passages faibles. Il est souhaitable, du point de vue de l'auditeur, d'abaisser les sons les plus forts vers le niveau dialogue et de relever les passages faibles également vers le niveau dialogue. Le gain des sons qui sont au niveau du dialogue parlé normal ne sera habituellement pas modifié.



La compression est effectivement appliquée au signal audio par le décodeur AC-3. Le signal audio codé contient l'intégralité de la plage dynamique. Le décodeur AC-3 peut (par action facultative de l'utilisateur) ignorer les valeurs de `dynrng` du flux binaire. On obtient ainsi une reproduction audio avec une plage dynamique intégrale. Le décodeur peut aussi, au choix de l'utilisateur, utiliser une fraction de la valeur de commande de `dynrng` et utiliser une fraction différente de valeurs positives et négatives. Le décodeur AC-3 peut ainsi reproduire un signal audio entièrement comprimé (comme prévu par la commande de compression du codeur AC-3), un signal audio de plage dynamique complète ou un signal audio de plage dynamique partiellement comprimée avec des rapports de compression différents pour les signaux de bas et de haut niveaux.

*Exemple:*

Une piste sonore de film de longue durée est codée selon la norme AC-3. Le mixage du programme d'origine se situe à un niveau de dialogue de  $-25$  dB. Les explosions atteignent un niveau de crête de  $0$  dB à pleine échelle. Certains sons sont prévus pour être entendus par tous les auditeurs à un niveau de  $50$  dB en dessous du niveau de dialogue (soit  $-75$  dB). Un signal de commande de compression (une séquence de valeurs de `dynrng`) est généré par le codeur AC-3. Pendant les portions du programme audio où le niveau audio est supérieur au niveau de dialogue, les valeurs de `dynrng` indiquent un gain négatif ou une réduction de gain. Pour les signaux à pleine échelle à  $0$  dB (les explosions les plus fortes), une réduction de  $-15$  dB est codée dans `dynrng`. Pour les signaux très faibles, un accroissement de gain de  $20$  dB est codé dans `dynrng`.

Un auditeur souhaite reproduire cette piste sonore à faible niveau de manière à ne pas déranger tout en désirant entendre la totalité du contenu sonore du programme d'origine. Le décodeur peut alors utiliser le mode de reproduction par défaut, ce qui signifie une compression totale. L'auditeur règle le niveau de dialogue à  $60$  dB de niveau de pression sonore. L'intensité des explosions est limitée à  $70$  dB (elles se trouvent à  $25$  dB au-dessus du niveau du dialogue, mais un gain de  $-15$  dB est appliqué) alors que les sons faibles sont reproduits à  $30$  dB de niveau de pression acoustique (un gain de  $20$  dB est appliqué à leur niveau d'origine de  $50$  dB au-dessous du niveau de dialogue). La plage dynamique reproduite est de  $70$  dB  $- 30$  dB =  $40$  dB.

La situation d'écoute change et l'auditeur souhaite porter le niveau de reproduction du dialogue à niveau de pression sonore à  $70$  dB, tout en limitant le niveau sonore du programme. Les sons les plus faibles peuvent être reproduits au même niveau que précédemment. L'auditeur donne au décodeur AC-3 l'instruction d'utiliser les mêmes valeurs de `dynrng` que précédemment pour celles qui indiquent une réduction de gain, mais de réduire par un facteur  $1/2$  celles qui indiquent un accroissement de gain. Les explosions sont toujours reproduites à  $10$  dB au-dessus du niveau de dialogue qui est maintenant de  $80$  dB. Les sons faibles sont rehaussés d'un niveau de  $20$  dB/2 =  $10$  dB. Ils sont maintenant reproduits à  $40$  dB au-dessous du niveau de dialogue soit avec  $30$  dB de niveau de pression sonore. La plage dynamique reproduite est maintenant de  $80$  dB  $- 30$  dB =  $50$  dB.

Un autre auditeur souhaite disposer de la plage dynamique complète du signal audio d'origine. Cet auditeur ajuste le niveau du dialogue reproduit à  $75$  dB de niveau de pression acoustique et donne au décodeur AC-3 l'instruction d'ignorer le signal de commande de plage dynamique. Pour cet auditeur, les sons faibles sont reproduits à  $25$  dB de niveau de pression sonore et les explosions atteignent  $100$  dB de niveau de pression sonore. La plage dynamique reproduite est de  $100$  dB  $- 25$  dB =  $75$  dB. Ceci est exactement ce qui est prévu par le producteur du programme d'origine.

Pour être efficace, la méthode de commande de plage dynamique doit être utilisée par tous les fournisseurs de programmes. Comme tous les diffuseurs souhaitent fournir des programmes sous la forme la plus utilisable par leurs auditeurs, ils appliqueront pratiquement tous une compression de plage dynamique pour tout programme audio de plage dynamique importante. Cette compression n'est réversible que si elle a été réalisée par la technique intégrée au codage AC-3. Si les diffuseurs utilisent le système de commande de plage dynamique intégré au codage AC-3, les auditeurs pourront agir dans une certaine mesure sur la plage dynamique qu'ils reproduisent. Les diffuseurs doivent être certains que les caractéristiques qu'ils introduisent dans leur codage AC-3 seront utilisées par défaut lors de l'écoute par les auditeurs. A cet effet, le décodeur AC-3 doit appliquer par défaut les caractéristiques de compression indiquées dans les valeurs de `dynrng` du flux binaire. Les décodeurs AC-3 peuvent en option permettre à l'auditeur de contrôler l'utilisation des valeurs de `dynrng`, de manière à lui permettre une sélection partielle ou totale de la reproduction de la plage dynamique.

### 7.7.1.2 Réalisation détaillée

Le champ `dynrng` du flux binaire AC-3 a une longueur de 8 bits. Dans le cas où le code `acmod` = 0 (mode 1 + 1 ou 2 canaux complètement indépendants) `dynrng` s'applique au premier canal (canal1) et `dynrng2` au deuxième canal (canal2). La description de `dynrng` donnée ci-dessous est identique à celle de `dynrng2` pour le canal2. La valeur de `dynrng` peut être présente dans tout bloc audio. Si la valeur n'est pas présente, celle du bloc précédent est réutilisée sauf pour le bloc 0. Si une valeur de `dynrng` n'est pas présente dans le bloc 0, une valeur de 0000 0000 devrait être utilisée.

Le bit le plus significatif de `dynrng` et de `dynrng2` est transmis en tête. Les trois premiers bits indiquent des modifications de gain par incréments de 6,02 dB pouvant être réalisées par décalage arithmétique. Les bits suivants indiquent des modifications de gain linéaires et nécessitent une multiplication sur 6 bits. Nous représenterons les champs de 3 et de 5 bits du gain `dynrng` par la notation suivante:

$$X_0 X_1 X_2 \cdot Y_3 Y_4 Y_5 Y_6 Y_7$$

La signification des valeurs de X est décrite de la manière la plus simple en considérant que X représente un entier avec signe pouvant prendre des valeurs de -4 à 3. Le gain indiqué par X est alors égal à  $(X + 1) * 6,02$  dB. Le détail des valeurs de X est donné dans le Tableau 43.

TABLEAU 43

Signification des 3 bits les plus significatifs de `dynrng`

$X_0$	$X_1$	$X_2$	Valeur entière	Gain indiqué (dB)	Décalages arithmétiques
0	1	1	3	+24,08	4 à gauche
0	1	0	2	+18,06	3 à gauche
0	0	1	1	+12,04	2 à gauche
0	0	0	0	+6,02	1 à gauche
1	1	1	-1	0	Aucun
1	1	0	-2	-6,02	1 à droite
1	0	1	-3	-12,04	2 à droite
1	0	0	-4	-18,06	3 à droite

La valeur de Y représente une variation linéaire du gain jusqu'à -6 dB. Y peut être considéré comme un entier fractionnaire sans signe commençant par 1, soit:  $0,1 Y_3 Y_4 Y_5 Y_6 Y_7$  (base 2). Y peut représenter des valeurs comprises entre  $0,111111_2$  (ou 63/64) et  $0,100000_2$  (ou 1/2). En conséquence, Y peut représenter des modifications de gain entre -0,14 dB et -6,02 dB.

La combinaison des valeurs de X et Y permet à `dynrng` d'indiquer des variations de gain entre  $24,08 - 0,14 = +23,94$  dB, et  $-18,06 - 6 = -24,06$  dB. Le code binaire 0000 0000 indique un gain (unité) de 0 dB.

### Compression partielle

L'interprétation de la valeur de `dynrng` peut être changée de manière à représenter une modification de gain qui est une fraction de la valeur d'origine. La modification du taux de compression à appliquer est déterminée en considérant `dynrng` comme un nombre fractionnaire avec signe:

$$X_0 \cdot X_1 X_2 Y_3 Y_4 Y_5 Y_6 Y_7$$

$X_0$  représentant le bit de signe et  $X_1 X_2 Y_3 Y_4 Y_5 Y_6 Y_7$  une fraction de 7 bits. Ce nombre fractionnaire avec signe de 8 bits peut être multiplié avec une fraction indiquant la fraction de la compression initiale devant être appliquée. Si la valeur est multipliée par 1/2, la plage de compression de  $\pm 24$  dB sera réduite à  $\pm 12$  dB. Après la mise à l'échelle par multiplication, le résultat de 8 bits est ensuite interprété dans sa forme d'origine  $X_0 X_1 X_2 \cdot Y_3 Y_4 Y_5 Y_6 Y_7$  et utilisée normalement.

### 7.7.2 Compression forte; `compr`, `compr2`

L'élément `compr` permet au fournisseur de programme (ou diffuseur) de mettre en œuvre une réduction importante de plage dynamique (compression forte) d'une manière assurant qu'un sous-mixage monophonique n'excède pas un niveau de crête déterminé. Un signal audio fortement comprimé peut être souhaitable dans certaines situations d'écoute telles que la diffusion de films dans des chambres d'hôtel ou pour un dispositif intégré dans un siège d'avion. La limitation des crêtes est utile quand, par exemple, un sous-mixage monophonique doit alimenter un modulateur radiofréquence et qu'on doit éviter toute surmodulation.

### 7.7.2.1 Présentation générale

Certains équipements décodant le flux binaire AC-3 déclarent le signal audio par une liaison à plage dynamique très réduite. C'est le cas, par exemple, d'un décodeur de signal de télévision qui doit moduler l'image et le son reçus sur un canal radioélectrique fournissant un signal utilisable par un récepteur de télévision bon marché. Il est alors nécessaire de limiter le niveau maximum de crête de sortie à une valeur donnée afin d'éviter une surmodulation. La plupart du temps, le signal de commande de plage dynamique, *dynrng*, apporte une réduction de gain appropriée, de sorte que le niveau absolu de crête est limité. Toutefois, comme le système de commande de plage dynamique a pour objet de réaliser une réduction subjectivement agréable des niveaux sonores élevés perçus, il n'y a pas de garantie que les crêtes instantanées seront limitées de manière à éviter une surmodulation.

Un second signal de commande, *compr*, (*compr2* pour canal2 dans le mode 1 + 1) peut être présent dans le flux binaire AC-3 afin de limiter le niveau du signal AC-3 décodé. Ce signal de commande devrait être présent dans tous les flux binaires qui sont destinés à être reçus, par exemple, sur un décodeur extérieur connecté à un téléviseur. Le signal de commande *compr* est comparable au signal de commande *dynrng* utilisé par le décodeur pour modifier le niveau audio reproduit. Le signal de commande *compr* a une plage de commande qui est le double de la plage de commande de *dynrng* ( $\pm 48$  dB au lieu de  $\pm 24$  dB) avec une résolution (0,5 dB contre 0,25 dB). En outre, comme le signal de commande *compr* se trouve dans l'information BSI, sa résolution dans le temps n'est que d'une trame AC-3 (32 ms) au lieu d'un bloc (5,3 ms).

Certains équipements qui exigent de limiter le niveau des crêtes du signal audio devraient utiliser *compr* au lieu de *dynrng* lorsque *compr* est présent dans l'information BSI. Comme l'utilisation de *dynrng* évite la plupart du temps la présence des crêtes élevées, il n'est pas nécessaire pour le codeur AC-3 d'insérer *compr* sauf occasionnellement, c'est-à-dire pendant les instants où l'utilisation de *dynrng* conduirait à un niveau de crête excessif. Si le décodeur a reçu l'instruction d'utiliser *compr* et si *compr* n'est pas présent dans une trame, le signal de commande *dynrng* doit être utilisé pour la trame concernée.

Pour certaines applications du codage AC-3, certains récepteurs peuvent avoir à reproduire une plage dynamique très réduite. Dans ce cas, le signal de commande *compr* peut être présent en permanence. L'utilisation de *compr* au lieu de *dynrng* peut permettre la reproduction du signal audio avec une plage dynamique très limitée. Ceci peut être utile, par exemple, pour la diffusion de signaux audio dans une chambre d'hôtel ou pour un dispositif intégré dans un siège d'avion.

### 7.7.2.2 Réalisation détaillée

Le champ *compr* du flux binaire AC-3 a une longueur de 8 bits. Dans le cas où *acmod* = 0 (mode 1 + 1 ou 2 canaux totalement indépendants) *compr* s'applique au premier canal (*canal1*) et *compr2* s'applique au deuxième canal (*canal2*). La description de *compr* est donnée ci-dessous (pour *canal1*) et le traitement de *compr2* pour le canal2 est identique.

Le bit le plus significatif est transmis en tête. Les quatre premiers bits indiquent des modifications de gain par incréments de 6,02 dB pouvant être réalisées par une opération de décalage arithmétique. Les 4 bits suivants indiquent des modifications de gain linéaires et nécessitent une multiplication sur 5 bits. Nous représenterons les deux champs de 4 bits de *compr* par la notation suivante:

$$X_0 X_1 X_2 X_3 \cdot Y_4 Y_5 Y_6 Y_7$$

Pour expliquer la signification des valeurs de X, le plus simple est de considérer que X représente un entier sur 4 bits avec signe dont la valeur est comprise entre -8 et +7 inclus. Le gain indiqué par X est alors  $(X + 1) \cdot 6,02$  dB. Le détail est donné dans le Tableau 44.

La valeur de Y représente une variation linéaire du gain jusqu'à -6 dB. Y peut être considéré comme un entier fractionnaire sans signe, avec une valeur de tête de 1, soit:  $0,1 Y_4 Y_5 Y_6 Y_7$  (base 2). Y peut représenter des valeurs entre  $0,11111_2$  (ou  $63/64$ ) et  $0,10000_2$  (ou  $1/2$ ). Y peut représenter des valeurs entre  $0,11111_2$  (ou  $31/32$ ) et  $0,10000_2$  (ou  $1/2$ ). En conséquence, Y peut représenter des modifications de gain comprises entre -0,28 dB et -6,02 dB.

La combinaison des valeurs de X et Y permet à *compr* d'indiquer des variations de gain comprises entre  $48,16 - 0,28 = +47,88$  dB et  $-42,14 - 6 = -48,14$  dB.

TABLEAU 44

## Signification des 4 bits les plus significatifs de compr

$X_0$	$X_1$	$X_2$	$X_3$	Valeur entière	Gain indiqué (dB)	Décalages arithmétiques
0	1	1	1	7	+48,16	8 à gauche
0	1	1	0	6	+42,14	7 à gauche
0	1	0	1	5	+36,12	6 à gauche
0	1	0	0	4	+30,10	5 à gauche
0	0	1	1	3	+24,08	4 à gauche
0	0	1	0	2	+18,06	3 à gauche
0	0	0	1	1	+12,04	2 à gauche
0	0	0	0	0	+6,02	1 à gauche
1	1	1	1	-1	0	Aucun
1	1	1	0	-2	-6,02	1 à droite
1	1	0	1	-3	-12,04	2 à droite
1	1	0	0	-4	-18,06	3 à droite
1	0	1	1	-5	-24,08	4 à droite
1	0	1	0	-6	-30,10	5 à droite
1	0	0	1	-7	-36,12	6 à droite
1	0	0	0	-8	-42,14	7 à droite

## 7.8 Sous-mixage

Dans beaucoup de systèmes de reproduction, le nombre de haut-parleurs ne correspond pas au nombre de canaux audio codés. Un sous-mixage est donc nécessaire pour reproduire le programme audio complet. La normalisation du sous-mixage est importante afin que les fournisseurs de programmes puissent savoir avec certitude comment leur programme sera reproduit sur des systèmes qui ne sont pas équipés du même nombre de haut-parleurs. Avec des équations de sous-mixage normalisées, les producteurs de programmes peuvent vérifier la façon dont les versions sous-mixées vont être perçues et faire les ajustements nécessaires afin que les résultats soient acceptables pour les auditeurs. Le fournisseur de programme peut utiliser les éléments syntactiques `cmixlev` et `smixlev` en vue de modifier l'équilibre relatif des canaux central et ambiophonique par rapport aux canaux gauche et droite.

Le sous-mixage du canal d'effets lfe est optionnel. Un sous-mixage idéal serait que le canal d'effets lfe reproduise un niveau acoustique de +10 dB par rapport aux canaux gauche et droit. Comme l'inclusion de ce canal est optionnelle, tout coefficient de sous-mixage peut être utilisé en pratique. On devrait prendre certaines précautions pour garantir que les haut-parleurs ne sont pas surchargés par le contenu basse fréquence à pleine échelle du canal d'effets lfe.

### 7.8.1 Procédure générale de sous-mixage

Le pseudo-code ci-dessous décrit la façon d'obtenir des coefficients de sous-mixage `downmix` non normalisés. Il peut être nécessaire, dans une réalisation concrète, de normaliser par la suite les coefficients `downmix` afin d'éviter tout risque de surcharge. La normalisation peut être réalisée en diminuant tous les coefficients `downmix` d'une manière égale de façon à ce que la somme des coefficients utilisés pour créer un canal de sortie unique n'exède jamais 1.

Pseudo-code

```

downmix()
{
  if (acmod == 0) /* mode 1+1, deux canaux mono indépendants */
  {
    if (output_nfront == 1) /* 1 haut-parleur avant (central) */
    {
      if (dualmode == Chan 1) /* sortie canal 1 demandée */
      {
        route left into center;
      }
      else if (dualmode == Chan 2) /* sortie canal 2 demandée */
      {
        route right into center;
      }
      else
      {
        mix left into center with -6 dB gain;
        mix right into center with -6 dB gain;
      }
    }
    else if (output_nfront == 2) /* 2 haut-parleurs avant (gauche, droite) */
    {
      if (dualmode == Stereo) /* sortie des deux canaux mono demandée */
      {
        route left into left;
        route right into right;
      }
      else if (dualmode == Chan 1)
      {
        mix left into left with -3 dB gain;
        mix left into right with -3 dB gain;
      }
      else if (dualmode == Chan 2)
      {
        mix right into left with -3 dB gain;
        mix right into right with -3 dB gain;
      }
      else /* somme mono des deux canaux mono demandée */
      {
        mix left into left with -6 dB gain;
        mix right into left with -6 dB gain;
        mix left into right with -6 dB gain;
        mix right into right with -6 dB gain;
      }
    }
  }
  else /* output_nfront==3 */
  {
    if (dualmode == Stereo)
    {
      route left into left;
      route right into right;
    }
    else if (dualmode == Chan 1)
    {
      route left into center;
    }
    else if (dualmode == Chan 2)
    {
      route right into center;
    }
    else
    {
      mix left into center with -6 dB gain;
      mix right into center with -6 dB gain;
    }
  }
}

```

Pseudo-code

```

else /* acmod > 0 */
{
  for i = { left, center, right, leftsur/monosur, rightsur }
  {
    if (exists(input_chan[i])) and (exists(output_chan[i]))
    {
      route input_chan[i] into output_chan[i];
    }
  }
  if (output_mode == 2/0 Dolby Surround compatible)
  /* sortie de la matrice codée à deux canaux demandée */
  {
    if (input_nfront != 2)
    {
      mix center into left with -3 dB gain;
      mix center into right with -3 dB gain;
    }
    if (input_nrear == 1)
    {
      mix -mono surround into left with -3 dB gain;
      mix mono surround into right with -3 dB gain;
    }
    else if (input_nrear == 2)
    {
      mix -left surround into left with -3 dB gain;
      mix -right surround into left with -3 dB gain;
      mix left surround into right with -3 dB gain;
      mix right surround into right with -3 dB gain;
    }
  }
}
else if (output_mode == 1/0) /* centre uniquement */
{
  if (input_nfront != 1)
  {
    mix left into center with -3 dB gain;
    mix right into center with -3 dB gain;
  }
  if (input_nfront == 3)
  {
    mix center into center using clef and -3 dB gain;
  }
  if (input_nrear == 1)
  {
    mix mono surround into center using slef and -3 dB gain;
  }
  else if (input_nrear == 2)
  {
    mix left surround into center using slef and -3 dB gain;
    mix right surround into center using slef and -3 dB gain;
  }
}
else /* plus que la sortie du centre exigée */
{
  if (output_nfront == 2)
  {
    if (input_nfront == 1)
    {
      mix center into left with -3 dB gain;
      mix center into right with -3 dB gain;
    }
    else if (input_nfront == 3)
    {
      mix center into left using clef;
      mix center into right using clef;
    }
  }
}

```

Pseudo-code

```

if (input_nrear == 1) /* canal ambiophonique codé unique */
{
    if (output_nrear == 0) /* pas de haut-parleurs ambiophoniques */
    {
        mix mono surround into left with slev and -3 dB gain;
        mix mono surround into right with slev and -3 dB gain;
    }
    else if (output_nrear == 2) /* deux canaux de haut-parleurs ambiophoniques */
    {
        mix mono srnd into left surround with -3 dB gain;
        mix mono srnd into right surround with -3 dB gain;
    }
}
else if (input_nrear == 2) /* deux canaux ambiophoniques codés */
{
    if (output_nrear == 0)
    {
        mix left surround into left using slev;
        mix right surround into right using slev;
    }
    else if (output_nrear == 1) .
    {
        mix left srnd into mono surround with -3 dB gain;
        mix right srnd into mono surround with -3 dB gain;
    }
}
}
}
}

```

Les coefficients réels utilisés pour le sous-mixage modifient le niveau absolu du canal central. Il faut tenir compte de ce phénomène lorsque le niveau de dialogue a été fixé avec un étalonnage absolu du niveau de pression acoustique.

### 7.8.2 Sous-mixage vers deux canaux

Soient L, C, R, L<sub>S</sub> et R<sub>S</sub> les 5 canaux séparés (gauche, centre, droite, ambiophonique gauche et ambiophonique droit) devant être sous-mixés à 2 canaux. Dans le cas d'un unique canal ambiophonique (modes n/1), S désigne le canal ambiophonique unique. Deux types de sous-mixage devraient être possibles: le sous-mixage vers une paire stéréo codée avec ambiophonie sous forme de matrice L<sub>t</sub>R<sub>t</sub> et le sous-mixage vers un signal stéréo normal L<sub>0</sub>R<sub>0</sub>. Le signal stéréo réduit (L<sub>0</sub>R<sub>0</sub> ou L<sub>t</sub>R<sub>t</sub>) peut ensuite être mixé vers un signal mono M par simple sommation des deux canaux. Si le sous-mixage L<sub>t</sub>R<sub>t</sub> est combiné sous forme mono, l'information ambiophonique est perdue. Le sous-mixage L<sub>0</sub>R<sub>0</sub> est préférable si on veut obtenir un signal mono. Les coefficients du sous-mixage doivent avoir une précision relative d'au moins ±0,25 dB.

Avant la normalisation nécessaire pour éviter tout dépassement de capacité, les équations générales de downmix 3/2 pour un signal stéréo L<sub>0</sub>R<sub>0</sub> sont les suivantes:

$$L_0 = 1,0 * L + cleav * C + slev * L_S;$$

$$R_0 = 1,0 * R + cleav * C + slev * R_S;$$

Si L<sub>0</sub>R<sub>0</sub> sont ensuite combinés pour obtenir une reproduction monophonique, l'équation finale du sous-mixage mono devient:

$$M = 1,0 * L + 2,0 * cleav * C + 1,0 * R + slev * L_S + slev * R_S;$$

Si un canal ambiophonique unique S est présent (mode 3/1) les équations de downmix sont les suivantes:

$$L_0 = 1,0 * L + cleav * C + 0,7 * slev * S;$$

$$R_0 = 1,0 * R + cleav * C + 0,7 * slev * S;$$

$$M = 1,0 * L + 2,0 * cleav * C + 1,0 * R + 1,4 * slev * S;$$

Les valeurs des coefficients cleav et slev sont données par les champs de bits cmixlev et surmixlev de l'information BSI, comme le montrent respectivement les Tableaux 4 et 5.

Si les champs de bits  $cmixlev$  et  $surmixlev$  indiquent l'état réservé (valeur 1 1), le décodeur devrait utiliser les valeurs du coefficient intermédiaires indiquées par le champ de bit de valeur 0 1. Si le canal central  $C$  est absent (mode 2/1 ou 2/2), les mêmes équations peuvent être utilisées en omettant le terme en  $C$ . Si les canaux ambiophoniques sont absents, les mêmes équations peuvent être utilisées en omettant les termes en  $L_s$ ,  $R_s$  ou  $S$ .

Avant la normalisation nécessaire pour éviter tout dépassement de capacité, les équations générales de  $downmix$  3/2 pour un signal stéréo  $L_tR_t$  sont les suivantes:

$$L_t = 1,0 * L + 0,707 * C - 0,707 * L_s - 0,707 * R_s;$$

$$R_t = 1,0 * R + 0,707 * C + 0,707 * L_s + 0,707 * R_s;$$

Si un unique canal ambiophonique,  $S$ , est présent (mode 3/1) ces équations deviennent:

$$L_t = 1,0 L + 0,707 C - 0,707 S;$$

$$R_t = 1,0 R + 0,707 C + 0,707 S;$$

Si le canal central est absent (mode 2/2 ou 2/1) le terme  $C$  est omis.

Les coefficients réels doivent être abaissés par mise à l'échelle de manière à éviter tout dépassement de capacité arithmétique lorsque tous les canaux constitutifs du signal sous-mixé sont à pleine échelle. Pour chaque mode de codage audio, un nombre différent de canaux participent au sous-mixage et de ce fait une mise à l'échelle différente peut être utilisée pour chaque cas afin d'éviter tout dépassement de la capacité. Pour simplifier, on peut utiliser la mise à l'échelle prévue pour le cas le plus défavorable. On minimise ainsi le nombre de coefficient nécessaire. Le cas le plus défavorable se présente lorsque les niveaux  $clev$  et  $slev$  sont tous deux égaux à 0,707. Dans le cas  $L_0R_0$ , la somme des coefficients non mis à l'échelle est de  $1 + 0,707 + 0,707 = 2,414$ , de sorte que tous les coefficients doivent être multipliés par  $1 / 2,414 = 0,4143$  (réduction d'échelle de 7,65 dB). Dans le cas  $L_tR_t$ , la somme des coefficients non mis à l'échelle est de  $1 + 0,707 + 0,707 + 0,707 = 3,121$ , de sorte que tous les coefficients doivent être multipliés par  $1 / 3,121 = 0,3204$  (réduction d'échelle de 9,89 dB). Les coefficients mis à l'échelle sont normalement convertis en valeurs binaires avec une longueur de mot limitée. Les coefficients de 6 bits donnés ci-dessous devraient avoir une précision suffisante.

Pour mettre en œuvre  $downmix$  vers deux canaux  $L_0R_0$  avec une mise à l'échelle de 0,453, il faut des valeurs de coefficients correspondant aux valeurs 1,0; 0,707; 0,596; 0,500 et 0,354.

TABLEAU 45  
Coefficients de  $downmix$   $L_0R_0$

Coefficient non normalisé	Coefficient normalisé	Coefficient quantifié avec 6 bits	Gain (dB)	Gain relatif (dB)	Erreur sur le coefficient (dB)
1,0	0,414	26/64	-7,8	0,0	-
0,707	0,293	18/64	-11,0	-3,2	-0,2
0,596	0,247	15/64	-12,6	-4,8	+0,3
0,500	0,207	13/64	-13,8	-6,0	0,0
0,354	0,147	9/64	-17,0	-9,2	-0,2

Pour mettre en œuvre  $downmix$  vers deux canaux  $L_tR_t$  avec une mise à l'échelle de 0,3204, il faut des valeurs de coefficients correspondant aux valeurs 1,0 et 0,707.

TABLEAU 46  
Coefficients de  $downmix$   $L_tR_t$

Coefficient non normalisé	Coefficient normalisé	Coefficient quantifié avec 6 bits	Gain (dB)	Gain relatif (dB)	Erreur sur le coefficient (dB)
1,0	0,3204	20/64	-10,1	0,0	-
0,707	0,2265	14/64	-13,20	-3,1	-0,10



S'il est nécessaire de mettre en œuvre un sous-mixage monophonique, une mise à l'échelle supplémentaire de 1/2 doit être appliquée aux coefficients  $L_0R_0$  afin d'éviter une surcharge du signal monophonique  $L_0 + R_0$ .

## 7.9 Equations de transformation et commutation des blocs

### 7.9.1 Présentation générale

Le choix de la longueur du bloc d'analyse est fondamental pour tout système de codage audio à transformation. Une longueur de transformée élevée est la mieux adaptée à des signaux d'entrée dont le spectre reste stationnaire ou ne varie que lentement avec le temps. Une longueur de transformée élevée donne une résolution en fréquence plus grande et améliore donc l'efficacité du codage de ces signaux. D'un autre côté, une taille de transformée plus courte possédant une plus grande résolution dans le temps est plus souhaitable pour des signaux qui varient rapidement dans le temps. Il est donc nécessaire d'établir un compromis entre résolution temporelle et résolution fréquentielle pour le choix d'une longueur de bloc de transformation.

La démarche traditionnelle pour résoudre ce dilemme consiste à choisir une taille de transformée unique qui fournit le meilleur compromis pour la qualité de codage aussi bien pour des signaux stationnaires que pour des signaux dynamiques. Le codage AC-3 utilise une démarche plus optimisée consistant à adapter la résolution fréquence/temps de la transformée en fonction des caractéristiques spectrales et temporelles du signal à traiter. Cette démarche est tout à fait comparable à ce que nous connaissons du comportement de l'ouïe humaine. Dans le codage de la transformée, l'adaptation se fait en commutant la longueur de bloc en fonction des caractéristiques du signal.

### 7.9.2 Technique

La procédure de commutation de bloc de la transformation AC-3 permet de choisir entre une longueur de bloc de 512 ou 256 échantillons, correspondant à une résolution temporelle de 10,7 ou 5,3 ms pour une fréquence d'échantillonnage de 48 kHz. La longueur normale de bloc est de 512 échantillons. Lorsqu'un bloc fenêtré normal subit une transformation, le résultat se présente sous la forme de 256 coefficients uniques de transformée dans le domaine fréquence. Des blocs plus courts sont construits en prenant le segment audio usuel correspondant à une fenêtre de 512 échantillons et en le divisant en deux segments de 256 échantillons chacun. La première moitié d'un bloc MDCT subit une transformation séparée, mais de la même manière que pour la seconde moitié. Chaque moitié de bloc produit 128 coefficients uniques et non nuls de la transformée qui représentent les fréquences de 0 à la moitié de la fréquence d'échantillonnage pour un total de 256. Ceci correspond au nombre de coefficients produits par un bloc unique de 512 échantillons, mais avec une résolution temporelle double. Les coefficients de transformée des deux demi-blocs sont entrelacés un à un afin de former un bloc unique de 256 valeurs. Ce bloc est quantifié et transmis à l'identique pour former un bloc long unique. Une procédure symétrique comparable est utilisée par le décodeur pendant la reconstitution du signal.

Les coefficients des deux transformées de longueur 256 arrivent au décodeur entrelacés selon leur rang. La séquence entrelacée contient le même nombre de coefficients de transformée que ceux générés par une transformation unique sur 512 échantillons. Le décodeur traite les séquences entrelacées de la même manière que les séquences non entrelacées, sauf au cours de la transformation inverse décrite ci-dessous.

Le codeur analyse les caractéristiques spectrales et/ou temporelles du signal d'entrée et sélectionne la valeur de bloc appropriée avant de transformer le signal audio du domaine temps vers le domaine fréquence. Cette analyse est faite uniquement par le codeur et peut donc être modifiée et améliorée sans altération de la base de décodeurs existante. Le flux binaire contient, par bloc de transformée et par canal, un code d'un bit ( $blksw[ch]$ ) transportant l'information de longueur ( $blksw[ch] = 0$  ou 1 respectivement pour 512 ou 256 échantillons). Le décodeur utilise cette information pour dépaqueter le flux binaire, reconstruire les données des mantisses et appliquer les équations de transformation inverses appropriées.

### 7.9.3 Réalisation du décodeur

La commutation de bloc de la transformation TDAC est réalisée dans le décodeur AC-3 en modifiant les équations normales de transformation directe et inverse de la transformation de longueur 256. La fenêtre et les tables de sinus/cosinus de la transformée de Fourier rapide (TFR) utilisées pour les blocs de 512 échantillons peuvent être réutilisées pour la transformation inverse des blocs de 256 échantillons; cependant les manipulations de multiplication complexe avant et après la TFR nécessitent un tableau additionnel de 128 valeurs pour le calcul de la transformée d'un bloc commuté.

Comme les tableaux d'entrée et de sortie ont pour,  $blksw[ch] = 1$ , une taille exactement moitié de taille correspondante pour  $blksw = 0$ , la taille en mémoire de la transformée inverse et des tampons associés est exactement la même avec ou sans commutateur de bloc.

Les modifications requises pour la transformation inverse des blocs de 256 échantillons sont les suivantes:

- Le tableau d'entrée contient 128 coefficients au lieu de 256.
- Les manipulations avant la TFR inverse (TFRI) utilisent une table de cosinus différente, ce qui nécessite une table supplémentaire à 128 entrées (64 cosinus et 64 sinus).
- La transformation TFRI complexe porte sur 64 points au lieu de 128. La même table TFR de cosinus peut être utilisée en ne prenant que les entrées paires.
- Les pointeurs d'entrée du traitement de fenêtre après la transformation TFRI sont initialisés avec des valeurs de départ différentes et sont calculés modulo 128 au lieu de modulo 256.

#### 7.9.4 Equations de la transformation

##### 7.9.4.1 Transformation IMDCT à 512 échantillons

La procédure ci-dessous décrit la technique utilisée pour calculer la transformée pour un bloc unique de données réelles de longueur  $N = 512$  mettant en œuvre une unique transformation TFRI complexe à  $N/4$  points avec des opérations simples de manipulation avant et après transformation. Les équations données sont celles de la transformation inverse utilisée quand l'indicateur `blksw` est positionné à 0 (indiquant l'absence d'un signal transitoire et des transformations sur 512 échantillons).

*Etape 1:* Définition des coefficients de la transformation MDCT =  $X[k]$ ,  $k = 0, 1, \dots, N/2-1$

*Etape 2:* Multiplication complexe avant TFRI

Calculer le résultat de la multiplication complexe sur  $N/8$  points  $Z[k]$ ,  $k = 0, 1, \dots, N/4-1$ :

Pseudo-code
<pre> for(k=0; k&lt;N/4; k++) {   /* Z[k] = (X[N/2-2*k-1] + j * X[2*k]) * (xcos1[k] + j * xsin1[k]); */   Z[k]=(X[N/2-2*k-1]*xcos1[k]-X[2*k]*xsin1[k])+j*(X[2*k]*xcos1[k]+X[N/2-2*k-1]*xsin1[k]); } </pre>

où:

$$\begin{aligned} \text{xcos1}[k] &= -\cos(2\pi * (8*k+1)/(8*N)); \\ \text{xsin1}[k] &= -\sin(2\pi * (8*k+1)/(8*N)); \end{aligned}$$

*Etape 3:* Transformation TFRI complexe

Calculer la transformée TFRI complexe de  $Z[k]$  donnant la séquence de valeurs complexes  $z[n]$ :

Pseudo-code
<pre> for(n=0; n&lt;N/4; n++) {   z[n] = 0;   for(k=0; k&lt;N/4; k++)   {     z[n] += Z[k] * (cos(8*pi*k*n/N) + j * sin(8*pi*k*n/N));   } } </pre>

*Etape 4:* Multiplication complexe après transformation TFRI

Calculer le résultat de la multiplication complexe sur  $N/4$  points  $y[n]$ ,  $n = 0, 1, \dots, N/4-1$ :

Pseudo-code
<pre> for(n=0; n&lt;N/4; n++) {   /* y[n] = z[n] * (xcos1[n] + j * xsin1[n]); */   y[n] = (zr[n] * xcos1[n] - zi[n] * xsin1[n]) + j * (zi[n] * xcos1[n] + zr[n] * xsin1[n]); } </pre>

où:

```

zr[n] = real(z[n]);
zi[n] = imag(z[n]);
xcos1[n] et xsin1[n] étant définis comme dans l'étape ci dessus.

```

*Etape 5:* Fenêtrage et désentrelacement

Calculer les échantillons dans la fenêtre du domaine temps  $x[n]$ :

Pseudo-code
<pre> for(n=0; n&lt;N/8; n++) {   x[2*n] = -yi[N/8+n] * w[2*n];   x[2*n+1] = yr[N/8-n-1] * w[2*n+1];   x[N/4+2*n] = -yr[n] * w[N/4+2*n];   x[N/4+2*n+1] = yi[N/4-n-1] * w[N/4+2*n+1];   x[N/2+2*n] = -yr[N/8+n] * w[N/2-2*n-1];   x[N/2+2*n+1] = yi[N/8-n-1] * w[N/2-2*n-2];   x[3*N/4+2*n] = yi[n] * w[N/4-2*n-1];   x[3*N/4+2*n+1] = -yr[N/4-n-1] * w[N/4-2*n-2]; } </pre>

où:

```

yr[n] = real(y[n]);
yi[n] = imag(y[n]);
w[n] est la séquence de fenêtre de la transformation (voir le Tableau 47).

```

*Etape 6:* Superposition et addition

La première moitié du bloc fenêtré est superposée à la deuxième moitié du bloc précédent pour produire les échantillons MIC (le facteur d'échelle de 2 compense la normalisation de plafond faite par le codeur).

Pseudo-code
<pre> for(n=0; n&lt;N/2; n++) {   pcm[n] = 2 * (x[n] + delay[n]);   delay[n] = x[N/2+n]; } </pre>

Il est à noter que le traitement arithmétique de l'étape de superposition/addition doit utiliser une arithmétique à saturation pour empêcher tout dépassement de capacité (bouclage). Comme le signal de sortie est constitué de la somme du signal d'origine et de l'erreur de codage, il est possible que le signal de sortie dépasse le niveau 100% même si le signal d'entrée est inférieur ou égal à 100%.

### 7.9.4.2 Transformées IMDCT à 256 échantillons

Les équations suivantes devraient être utilisées pour calculer les transformées inverses dans le cas où  $\text{blksw}=1$ , indiquant la présence d'un signal transitoire et de deux transformées à 256 échantillons (la valeur de  $N$  ci-dessous reste égale à 512).

*Etape 1:* Définition des coefficients de la transformation MDCT =  $X[k]$ ,  $k = 0, 1, \dots, N/2$

Pseudo-code
<pre> for(k=0; k&lt;N/4; k++) {     X1[k] = X[2*k];     X2[k] = X[2*k+1]; } </pre>

*Etape 2:* Multiplication complexe avant TFRI

Calculer les résultats de la multiplication complexe sur  $N/8$  points  $Z1[k]$  and  $Z2[k]$ ,  $k = 0, 1, \dots, N/8-1$ .

Pseudo-code
<pre> for(k=0; k&lt;N/8; k++) {     /* Z1[k] = (X1[N/4-2*k-1] + j * X1[2*k]) * (xcos2[k] + j * xsin2[k]); */     Z1[k]=(X1[N/4-2*k-1]*xcos2[k]-X1[2*k]*xsin2[k])+j*(X1[2*k]*xcos2[k]+X1[N/4-2*k-1]*xsin2[k]);     /* Z2[k] = (X2[N/4-2*k-1] + j * X2[2*k]) * (xcos2[k] + j * xsin2[k]); */     Z2[k]=(X2[N/4-2*k-1]*xcos2[k]-X2[2*k]*xsin2[k])+j*(X2[2*k]*xcos2[k]+X2[N/4-2*k-1]*xsin2[k]); } </pre>

où:

$$\text{xcos2}[k] = -\cos(2\pi*(8*k+1)/(4*N)), \quad \text{xsin2}(k) = -\sin(2\pi*(8*k+1)/(4*N))$$

*Etape 3:* TFRI complexe

Calculer les transformées TFRI complexes à  $N/8$  points de  $Z1[k]$  et  $Z2[k]$  donnant la séquence de valeurs complexes  $z1[n]$  et  $z2[n]$ .

Pseudo-code
<pre> for(n=0; n&lt;N/8; n++) {     z1[n] = 0.;     z2[n] = 0.;     for(k=0; k&lt;N/8; k++)     {         z1[n] += Z1[k] * (cos(16*pi*k*n/N) + j * sin(16*pi*k*n/N));         z2[n] += Z2[k] * (cos(16*pi*k*n/N) + j * sin(16*pi*k*n/N));     } } </pre>

*Etape 4:* Multiplication complexe après TFRI

Calculer les résultats de la multiplication complexe sur  $N/8$  points  $y1[n]$  et  $y2[n]$ ,  $n = 0, 1, \dots, N/8-1$ .

Pseudo-code
<pre> for(n=0; n&lt;N/8; n++) {   /* y1[n] = z1[n] * (xcos2[n] + j * xsin2[n]); */   y1[n] = (zr1[n] * xcos2[n] - zi1[n] * xsin2[n]) + j * (zi1[n] * xcos2[n] + zr1[n] * xsin2[n]);   /* y2[n] = z2[n] * (xcos2[n] + j * xsin2[n]); */   y2[n] = (zr2[n] * xcos2[n] - zi2[n] * xsin2[n]) + j * (zi2[n] * xcos2[n] + zr2[n] * xsin2[n]); } </pre>

où:

```

zr1[n] = real(z1[n]);
zi1[n] = imag(z1[n]);
zr2[n] = real(z2[n]);
zi2[n] = imag(z2[n]);
xcos2[n] et xsin2[n] étant définis comme dans l'étape 2 ci-dessus.

```

*Etape 5:* Fenêtrage et désentrelacement

Calcul des échantillons du domaine temps fenêtré  $x[n]$

Pseudo-code
<pre> for(n=0; n&lt;N/8; n++) {   x[2*n] = -yi1[n] * w[2*n];   x[2*n+1] = yr1[N/8-n-1] * w[2*n+1];   x[N/4+2*n] = -yr1[n] * w[N/4+2*n];   x[N/4+2*n+1] = yi1[N/8-n-1] * w[N/4+2*n+1];   x[N/2+2*n] = -yr2[n] * w[N/2-2*n-1];   x[N/2+2*n+1] = yi2[N/8-n-1] * w[N/2-2*n-2];   x[3N/4+2*n] = yi2[n] * w[N/4-2*n-1];   x[3N/4+2*n+1] = -yr2[N/8-n-1] * w[N/4-2*n-2]; } </pre>

où:

```

yr1[n] = real(y1[n]);
yi1[n] = imag(y1[n]);
yr2[n] = real(y2[n]);
yi2[n] = imag(y2[n]);
w[n] est la séquence de fenêtre de la transformation (voir le Tableau 47).

```

*Etape 6:* Etape de superposition partielle et d'addition

La première moitié du bloc sélectionné par la fenêtre est superposée à la deuxième moitié du bloc précédent pour produire les échantillons MIC (le facteur d'échelle de 2 compense la normalisation de plafond faite par le codeur).

Pseudo-code
<pre> for(n=0; n&lt;N/2; n++) {   pcm[n] = 2 * (x[n] + delay[n]);   delay[n] = x[N/2+n]; } </pre>

Il est à noter que le traitement arithmétique de l'étape de superposition addition doit utiliser une arithmétique à saturation pour prévenir le bouclage par débordement. Comme le signal de sortie est constitué de la somme du signal d'origine et de l'erreur de codage, il est possible que le signal de sortie dépasse un niveau de 100% même si le signal d'entrée est inférieur ou égal à 100%.

TABLEAU 47

Séquence de fenêtre de transformée ( $w[addr]$ ),  
avec  $addr = (10 * A) + B$

	B = 0	B = 1	B = 2	B = 3	B = 4	B = 5	B = 6	B = 7	B = 8	B = 9
A = 0	0,00014	0,00024	0,00037	0,00051	0,00067	0,00086	0,00107	0,00130	0,00157	0,00187
A = 1	0,00220	0,00256	0,00297	0,00341	0,00390	0,00443	0,00501	0,00564	0,00632	0,00706
A = 2	0,00785	0,00871	0,00962	0,01061	0,01166	0,01279	0,01399	0,01526	0,01662	0,01806
A = 3	0,01959	0,02121	0,02292	0,02472	0,02662	0,02863	0,03073	0,03294	0,03527	0,03770
A = 4	0,04025	0,04292	0,04571	0,04862	0,05165	0,05481	0,05810	0,06153	0,06508	0,06878
A = 5	0,07261	0,07658	0,08069	0,08495	0,08935	0,09389	0,09859	0,10343	0,10842	0,11356
A = 6	0,11885	0,12429	0,12988	0,13563	0,14152	0,14757	0,15376	0,16011	0,16661	0,17325
A = 7	0,18005	0,18699	0,19407	0,20130	0,20867	0,21618	0,22382	0,23161	0,23952	0,24757
A = 8	0,25574	0,26404	0,27246	0,28100	0,28965	0,29841	0,30729	0,31626	0,32533	0,33450
A = 9	0,34376	0,35311	0,36253	0,37204	0,38161	0,39126	0,40096	0,41072	0,42054	0,43040
A = 10	0,44030	0,45023	0,46020	0,47019	0,48020	0,49022	0,50025	0,51028	0,52031	0,53033
A = 11	0,54033	0,55031	0,56026	0,57019	0,58007	0,58991	0,59970	0,60944	0,61912	0,62873
A = 12	0,63827	0,64774	0,65713	0,66643	0,67564	0,68476	0,69377	0,70269	0,71150	0,72019
A = 13	0,72877	0,73723	0,74557	0,75378	0,76186	0,76981	0,77762	0,78530	0,79283	0,80022
A = 14	0,80747	0,81457	0,82151	0,82831	0,83496	0,84145	0,84779	0,85398	0,86001	0,86588
A = 15	0,87160	0,87716	0,88257	0,88782	0,89291	0,89785	0,90264	0,90728	0,91176	0,91610
A = 16	0,92028	0,92432	0,92822	0,93197	0,93558	0,93906	0,94240	0,94560	0,94867	0,95162
A = 17	0,95444	0,95713	0,95971	0,96217	0,96451	0,96674	0,96887	0,97089	0,97281	0,97463
A = 18	0,97635	0,97799	0,97953	0,98099	0,98236	0,98366	0,98488	0,98602	0,98710	0,98811
A = 19	0,98905	0,98994	0,99076	0,99153	0,99225	0,99291	0,99353	0,99411	0,99464	0,99513
A = 20	0,99558	0,99600	0,99639	0,99674	0,99706	0,99736	0,99763	0,99788	0,99811	0,99831
A = 21	0,99850	0,99867	0,99882	0,99895	0,99908	0,99919	0,99929	0,99938	0,99946	0,99953
A = 22	0,99959	0,99965	0,99969	0,99974	0,99978	0,99981	0,99984	0,99986	0,99988	0,99990
A = 23	0,99992	0,99993	0,99994	0,99995	0,99996	0,99997	0,99998	0,99998	0,99998	0,99999
A = 24	0,99999	0,99999	0,99999	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000
A = 25	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000				

### 7.9.5 Code des plages de gain des canaux

Lorsque le niveau de signal est faible, la plage dynamique du signal audio décodé est généralement limitée par la longueur de mot utilisée dans le calcul de la transformée. L'utilisation d'une longueur de mot plus élevée améliore la plage dynamique mais augmente le coût, car la longueur de mot des unités arithmétiques et de la mémoire vive en service doit également être augmentée. Pour réduire la longueur de mot lors du calcul de la transformée, un élément syntaxique `gainrng[ch]` est inclus dans le flux binaire AC-3. Cet élément à deux bits existe pour chaque bloc codé et pour chaque canal.

L'élément `gainrng` est une valeur comprise entre 0 et 3. La valeur est une indication du niveau d'échantillon maximal à l'intérieur du bloc codé. Chaque bloc représente 256 nouveaux échantillons audio et 256 anciens échantillons audio. Avant l'application de la fenêtre à 512 points, la valeur maximale absolue des 512 valeurs MIC est déterminée. Sur la base de la **valeur maximale à l'intérieur du bloc**, la valeur de l'élément `gainrng` est déterminée comme suit:

Valeur maximale absolue (max)	<code>gainrng</code>
$\text{max} \geq 0,5$	0
$0,5 > \text{max} \geq 0,25$	1
$0,25 > \text{max} \geq 0,125$	2
$0,125 > \text{max}$	3

Si le codeur ne procède pas à l'identification de la valeur maximale absolue à l'intérieur de chaque bloc, la valeur de l'élément `gainrng` doit être mise à 0.

Le décodeur peut utiliser la valeur de `gainrng` pour calibrer les coefficients de transformation avant la transformation et pour mettre à l'échelle les valeurs après la transformation. Au moyen d'une conception judicieuse, le processus de mise à l'échelle après la transformation peut être exécuté juste à l'étage de sortie de l'équipement MIC en prévoyant un registre de sortie à 16 bits assurant une plage dynamique audiofréquence de 18 bits.

## 7.10 Détection d'erreur

Il existe plusieurs manières de déterminer, à partir des données AC-3, si des erreurs sont présentes dans une trame de données. Le décodeur peut être informé par le système de transport qui a fourni les données. L'intégrité des données peut être vérifiée en utilisant les codes CRC intégrés. En outre, quelques tests simples de cohérence des données reçues peuvent indiquer la présence d'erreurs. L'utilisateur peut définir la stratégie du décodeur en cas de détection d'erreurs. Les réponses possibles incluent le mode muet, la répétition de bloc ou la répétition de trame. La présente Norme ne spécifie pas de niveau de la détection d'erreur à appliquer ni le comportement en cas d'erreur mais laisse ces décisions au responsable des applications et des réalisations.

### 7.10.1 Vérification du CRC

Chaque trame AC-3 contient deux mots de code de CRC. `crc1` est le deuxième mot de 16 bits dans la trame et suit immédiatement le mot de synchronisation. `crc2` est le dernier mot de 16 bits de la trame et précède immédiatement le mot de synchronisation de la trame suivante. `crc1` couvre les premiers 5/8 de la trame à l'exception du mot de synchronisation. Les derniers 3/8 sont vérifiés par `crc2` qui couvre en plus la totalité de la trame à l'exception du mot de synchronisation. Le décodage des mots de synchronisation permet la détection d'erreurs.

Chacun des codes CRC de 16 bits est produit à l'aide du polynôme générateur suivant:  $x^{16} + x^{15} + x^2 + 1$ .

Les 5/8 d'une trame sont donnés par le Tableau 48 et peuvent être calculés au moyen des formules suivantes:

$$5/8\_framesize = \text{truncate}(\text{framesize} \div 2) + \text{truncate}(\text{framesize} \div 8);$$

ou

$$5/8\_framesize = (\text{int}) (\text{framesize} \gg 1) + (\text{int}) (\text{framesize} \gg 3);$$

la taille de trame étant calculée en nombre de mots de 16 bits. Le Tableau 48 donne les valeurs des 5/8 de la taille de trame en fonction du flux binaire AC-3 et de la fréquence d'échantillonnage audio.

TABLEAU 48

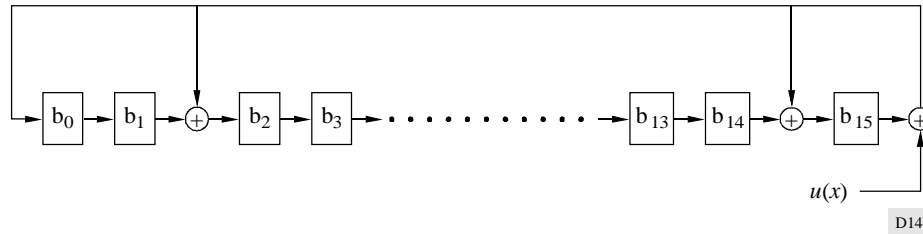
**5/8\_framesize; nombre de mots dans les premiers 5/8 de la trame**

Code de taille frmsizecod	Taux binaire nominal (kbit/s)	5/8_framesize $f_s = 32$ kHz	5/8_framesize $f_s = 44,1$ kHz	5/8_framesize $f_s = 48$ kHz
000000 (0)	32	60	42	40
000001 (0)	32	60	43	40
000010 (1)	40	75	53	50
000011 (1)	40	75	55	50
000100 (2)	48	90	65	60
000101 (2)	48	90	65	60
000110 (3)	56	105	75	70
000111 (3)	56	105	76	70
001000 (4)	64	120	86	80
001001 (4)	64	120	87	80
001010 (5)	80	150	108	100
001011 (5)	80	150	108	100
001100 (6)	96	180	130	120
001101 (6)	96	180	130	120
001110 (7)	112	210	151	140
001111 (7)	112	210	152	140
010000 (8)	128	240	173	160
010001 (8)	128	240	173	160
010010 (9)	160	300	217	200
010011 (9)	160	300	217	200
010100 (10)	192	360	260	240
010101 (10)	192	360	261	240
010110 (11)	224	420	303	280
010111 (11)	224	420	305	280
011000 (12)	256	480	347	320
011001 (12)	256	480	348	320
011010 (13)	320	600	435	400
011011 (13)	320	600	435	400
011100 (14)	384	720	521	480
011101 (14)	384	720	522	480
011110 (15)	448	840	608	560
011111 (15)	448	840	610	560
100000 (16)	512	960	696	640
100001 (16)	512	960	696	640
100010 (17)	576	1080	782	720
100011 (17)	576	1080	783	720
100100 (18)	640	1200	870	800
100101 (18)	640	1200	871	800



Le calcul du code de CRC peut être réalisé de plusieurs manières usuelles. On peut utiliser dans la pratique un registre à décalage linéaire à rétroaction (LFSR). La Fig. 14 donne un exemple de registre LFSR pour le polynôme générateur ci-dessous.

FIGURE 14



Pour effectuer la vérification du code de CRC à l'aide du circuit ci-dessus, il faut initialiser tous les registres à 0, puis décaler en série les bits du flux binaire AC-3 à travers le circuit dans l'ordre dans lequel ils apparaissent dans le flux binaire. Le mot de synchronisation n'est couvert par aucun des codes CRC (mais inclus dans l'indication 5/8\_framesize) et ne doit donc pas être pris en compte dans le calcul du CRC. *crc1* est considéré comme valide si le registre ci-dessus ne contient que des zéros après la réception par décalage des premiers 5/8 de la trame. Si le calcul se poursuit jusqu'à ce que toutes les données aient traversé le registre, et si la valeur est à nouveau nulle, alors *crc2* est considéré comme valide. Certains décodeurs peuvent faire le choix de vérifier uniquement *crc2* et de ne pas vérifier *crc1* aux 5/8 de la trame. Si *crc1* n'est pas valide, il est possible de remettre les registres à 0 et de vérifier *crc2*. Si le test de *crc2* est alors correct, il est probable que les derniers 3/8 de la trame ne contiennent pas d'erreur. Cette procédure n'est toutefois pas d'une grande utilité, étant donné que si des erreurs sont présentes dans les premiers 5/8 de la trame, il n'est pas possible de décoder un quelconque signal audio à partir de la trame, même si les derniers 3/8 sont sans erreur.

Il est à noter que *crc1* généré par les codeurs est tel que le calcul de CRC donne un résultat 0 aux 5/8 de la trame. Il ne s'agit *pas* de la valeur obtenue en calculant le code CRC des premiers 5/8 de la trame en utilisant le polynôme générateur ci-dessus. C'est pourquoi les décodeurs ne doivent pas essayer, pour sauvegarder *crc1*, de calculer le CRC des premiers 5/8 de la trame et de le comparer à *crc1*.

Les contraintes syntaxiques sur la taille des blocs appliquées par les codeurs garantissent que les blocs 0 et 1 sont totalement couverts par *crc1*. En conséquence les décodeurs peuvent immédiatement commencer à traiter le bloc 0 lorsque les 5/8 de la trame sont atteints, ce qui peut permettre, dans certaines applications, l'utilisation de tampons d'entrée plus réduits. Des décodeurs capables de stocker une trame entière peuvent faire le choix de ne traiter que *crc2*. Ces décodeurs ne commencent à traiter le bloc 0 d'une trame que lorsque toute la trame est reçue.

### 7.10.2 Vérification de l'homogénéité du flux binaire

Il est toujours possible qu'une trame AC-3 contenant une information de synchronisation valide et des codes CRC valides ne puisse être décodée. Cette situation peut se présenter si la trame contient une erreur telle que le mot de code de CRC reste valide ou en cas d'une erreur du codeur. Une protection contre de telles erreurs consiste à faire certaines détections d'erreur dans le décodeur AC-3 et dans l'analyseur du flux binaire. Malgré l'efficacité du codage, il existe certaines redondances propres au flux binaire AC-3. Si le flux binaire AC-3 contient des erreurs, un certain nombre de structures syntaxiques incorrectes ont une probabilité de se manifester. La vérification de ces structures incorrectes permet de détecter un grand nombre de conditions d'erreurs importantes.

La liste ci-dessous donne des conditions connues d'erreurs du flux binaire. Il peut être important, pour certaines réalisations, que le décodeur puisse traiter ces erreurs avec une certaine tolérance. D'une manière spécifique, dans certains décodeurs, il faut faire en sorte que ces erreurs n'entraînent pas l'écrasement de zones de mémoire réservées par des données invalides et n'entraînent pas des retards dans les traitements avec des compteurs d'itération incorrects. Une reproduction sonore incorrecte peut être acceptable tant que la stabilité du système est préservée.

- 1) (blknum == 0) &&  
(cplstre == 0);
- 2) (cplinu == 1) &&  
(no channels in coupling);
- 3) (cplinu == 1) &&  
(cplbegf > (cplendf+2));

- 4) (cplinu == 1) &&  
((blknum == 0) || (previous cplinu == 0)) &&  
(chincpl[n] == 1) &&  
(cplcoe[n] == 0);
- 5) (blknum == 0) &&  
(acmod == 2) &&  
(rematstr == 0);
- 6) (cplinu == 1) &&  
((blknum == 0) || (previous cplinu == 0)) &&  
(cplexpstr == 0);
- 7) (cplinu == 1) &&  
((cplbegf != previous cplbegf) || (cplendf != previous cplendf)) &&  
(cplexpstr == 0);
- 8) (blknum == 0) &&  
(chexpstr[n] == 0);
- 9) (cplinu == 1) &&  
(cplbegf != previous cplbegf) &&  
(chincpl[n] == 1) &&  
(chexpstr[n] == 0);
- 10) (blknum == 0) &&  
(lfeon == 1) &&  
(lfeexpstr == 0);
- 11) (chincpl[n] == 0) &&  
(chbwcod[n] > 60);
- 12) (blknum == 0) &&  
(baie == 0);
- 13) (blknum == 0) &&  
(snroffste == 0);
- 14) (blknum == 0) &&  
(cplinu == 1) &&  
(cplleake == 0);
- 15) (cplinu == 1) &&  
(taille décompressée de l'affectation de bits delta cpl > 50);
- 16) taille décompressée de l'affectation de bits delta[n] > 50;
- 17) valeur > 124 pour le codage composite d'exposant de 5;
- 18) valeur > 26 pour le codage composite de mantisse de niveau 3;
- 19) valeur > 124 pour le codage composite de mantisse de niveau 5;
- 20) valeur > 120 pour le codage composite de mantisse de niveau 11;
- 21) le dépaquetage du flux binaire se poursuit au-delà de la fin de la trame.

Il convient de noter que ces conditions (telles celles qui correspondent aux numéros 17 à 20) ne peuvent être testées qu'à de faibles niveaux dans le logiciel du décodeur, ce qui se traduit par des conséquences qui peuvent être importantes en terme de MIPS. Tant que ces conditions n'affectent pas la stabilité du système, il n'est pas nécessaire de prendre des précautions particulières.

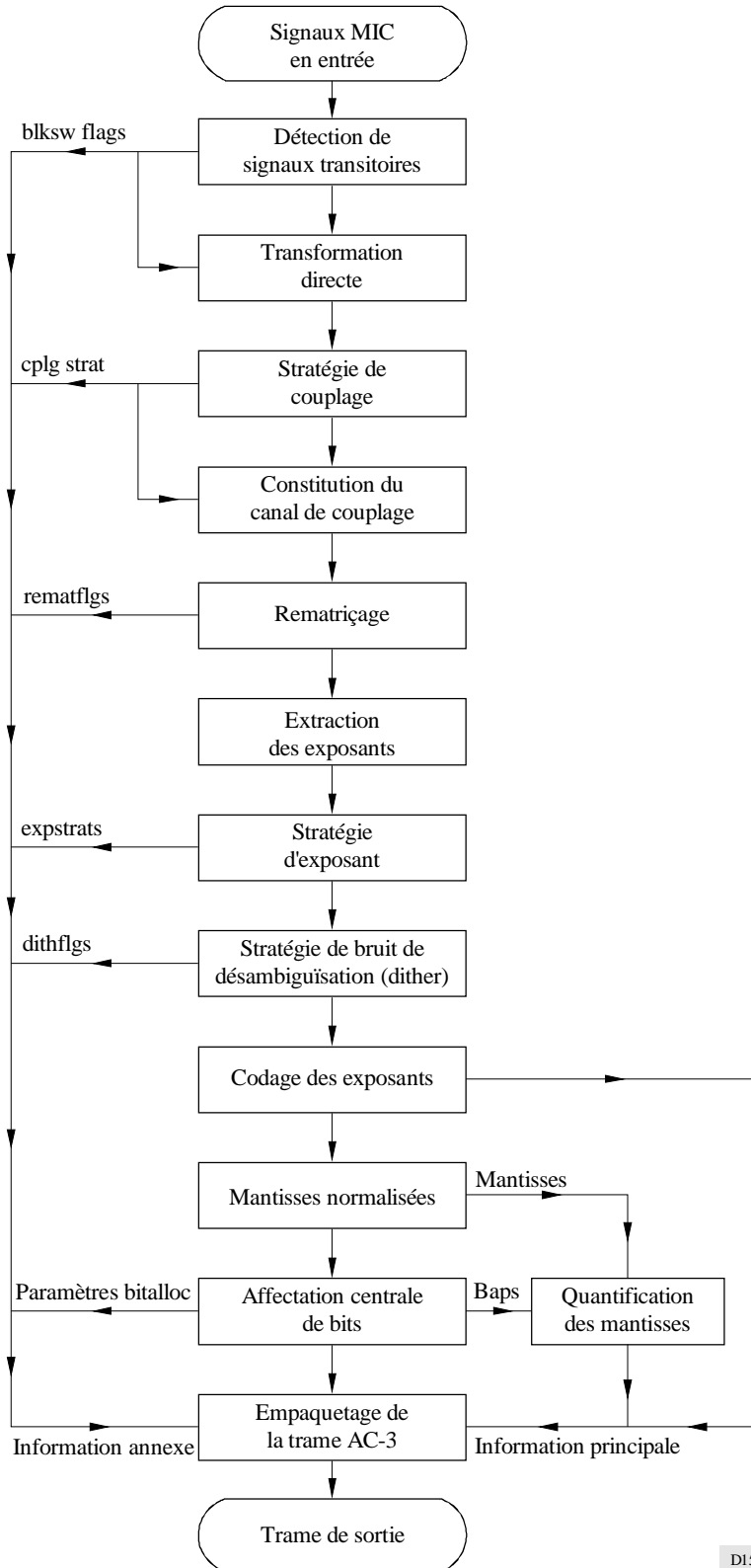
## 8 Codage du flux binaire AC-3

### 8.1 Introduction

Le présent paragraphe contient des indications sur le codage AC-3. Comme AC-3 est spécifié par la syntaxe et le traitement assuré par le décodeur, le codeur n'est pas spécifié d'une manière précise. La seule exigence normative concernant le codeur est que le flux binaire élémentaire de sortie respecte la syntaxe AC-3. Des codeurs de niveau de complexité divers peuvent être produits. Les codeurs les plus évolués peuvent présenter des performances audio supérieures ou permettre de tolérer une exploitation à des taux binaires plus faibles. Les performances des codeurs sont appelées à s'améliorer avec le temps. Tous les décodeurs bénéficieront des progrès réalisés pour les codeurs. Le codeur décrit dans

le présent paragraphe est élémentaire mais présente néanmoins de bonnes caractéristiques. La description qui suit indique plusieurs voies potentielles d'amélioration. La Fig. 15 donne un diagramme de fluence du processus de codage.

FIGURE 15  
Diagramme de fluence du processus de codage



## 8.2 Résumé du processus de codage

### 8.2.1 Signaux MIC en entrée

#### 8.2.1.1 Longueur du mot en entrée

Le codeur AC-3 accepte le signal audio sous la forme de mots MIC. La plage dynamique interne AC-3 permet d'utiliser des longueurs de mots en entrée allant jusqu'à 24 bits.

#### 8.2.1.2 Débit des échantillons en entrée

Le débit des échantillons en entrée doit être calé sur le débit binaire de sortie, de sorte que chaque trame de synchronisation AC-3 contienne 1 536 échantillons de signal audio. Si l'information audio d'entrée est disponible dans un format MIC avec une fréquence d'échantillonnage différente de celle requise, une conversion de fréquence d'échantillonnage doit alors être effectuée.

#### 8.2.1.3 Filtrage en entrée

Les canaux individuels en entrée peuvent subir un filtrage passe-haut. La suppression des composantes continues des signaux peut permettre un codage plus efficace car le débit binaire n'est pas utilisé pour coder les composantes continues. Il existe cependant un risque que des signaux n'atteignant pas le niveau MIC de 100% avant le filtrage passe-haut dépassent le niveau 100% après filtrage et doivent être écrêtés. Un codeur normal devrait effectuer un filtrage passe-haut des signaux d'entrée avec un filtre à pôle unique à 3 Hz.

Le canal d'effets lfe devrait subir un filtrage passe-bas à 120 Hz. Un codeur normal devrait filtrer le canal d'effets lfe au moyen d'un filtre elliptique d'ordre 8 ayant une fréquence de coupure de 120 Hz.

## 8.2.2 Détection de signaux transitoires

Une détection des transitoires est effectuée sur les canaux à largeur de bande complète afin de décider du moment de commutation vers des blocs audio de longueur courte dans le but d'améliorer le résultat avant écho. Les versions des signaux, filtrées par un passe-haut, sont analysées afin de détecter un accroissement d'énergie entre un segment de temps de sous-bloc et le suivant. Les sous-blocs sont analysés à différentes échelles de temps. Si une transitoire est détectée dans un canal pendant la seconde moitié d'un bloc audio, ce canal se commute pour utiliser un bloc court. Un canal qui a commuté sa longueur de bloc utilise la stratégie d'exposant D45.

Le détecteur de transitoires est utilisé pour déterminer à quel moment commuter d'un bloc de transformée de longueur 512 vers un bloc de transformée de longueur 256. Il traite 512 échantillons pour chaque bloc audio. Cette procédure est exécutée en deux passages, chaque passage traitant 256 échantillons. La détection de signal transitoire est divisée en quatre étapes:

*Etape 1:* un filtrage passe-haut,

*Etape 2:* une segmentation du bloc en sous-multiples,

*Etape 3:* une détection d'amplitude crête au sein de chaque segment de sous-bloc, et

*Etape 4:* une détection de seuil.

Le détecteur de transitoires positionne un indicateur `blksw[ch]` pour chaque canal de largeur de bande complète qui indique, lorsqu'il est positionné en «un», la présence d'une transitoire dans la seconde moitié du bloc d'entrée du canal correspondant.

*Etape 1: Filtrage passe-haut:* le filtre passe-haut est réalisé par un filtre II IIR de forme biquadratique directe en cascade ayant une fréquence de coupure de 8 kHz.

*Etape 2: Segmentation du bloc:* le bloc de 256 échantillons ayant subi un filtrage passe-haut est segmenté en un arbre de niveaux hiérarchiques dans lequel le niveau 1 représente le bloc de longueur 256, le niveau 2 est constitué de deux segments de longueur 128 et le niveau 3 de quatre segments de longueur 64.

*Etape 3: Détection des crêtes:* l'échantillon de plus grande amplitude est identifié dans chaque segment à tous les niveaux d'arbre hiérarchique. Les crêtes d'un niveau donné sont déterminées comme suit:

$$P[j][k] = \max(x(n))$$

$$\text{for } n = (512 \times (k-1) / 2j), (512 \times (k-1) / 2j) + 1, \dots, (512 \times k / 2j) - 1$$

$$\text{and } k = 1, \dots, 2^j - 1;$$

où:

$x(n)$  = le nième échantillon dans le bloc de longueur 256

$j$  = 1, 2, 3 est le numéro de la hiérarchie de blocs

$k$  = le rang du segment au sein du niveau  $j$

Il est à noter que  $P[j][0]$ , (c'est-à-dire pour  $k = 0$ ) est défini comme étant la crête du dernier segment du niveau  $j$  de l'arbre évalué immédiatement avant l'arbre en cours. Par exemple,  $P[3][4]$  de l'arbre précédent est  $P[3][0]$  de l'arbre en cours.

*Etape 4: Comparaison de seuil:* le comparateur de niveau vérifie d'abord s'il existe un niveau de signal important dans le bloc en cours. Par comparaison de la valeur de crête globale  $P[1][1]$  du bloc courant avec un «seuil de silence». Si  $P[1][1]$  est en dessous de ce seuil, un bloc long est forcé. La valeur du seuil de silence est de 100/32768. L'action suivante du comparateur examine à chaque niveau les pics relatifs de segments adjacents. Un indicateur est positionné, indiquant un signal transitoire dans le bloc courant de longueur 256, si le rapport entre les pics de n'importe quels segments adjacents d'un niveau donné excède un seuil prédéfini pour ce niveau. Les rapports sont comparés comme suit:

$$\text{mag}(P[j][k]) \times T[j] > \text{mag}(P[j][(k-1)])$$

où:

$T[j]$  donnant les seuils prédéfinis pour le niveau  $j$ , avec les valeurs:

$$T[1] = 0,1$$

$$T[2] = 0,075$$

$$T[3] = 0,05$$

Si l'inégalité est vraie pour deux segments quelconques de n'importe quel niveau, une indication de signal transitoire est faite pour la première moitié du bloc d'entrée de longueur 512. Le deuxième passage de ce traitement détermine la présence de signaux transitoires dans la deuxième moitié du bloc d'entrée de longueur 512.

### 8.2.3 Transformation directe

#### 8.2.3.1 Fenêtrage

Le bloc audio est multiplié par une fonction fenêtre afin de réduire les effets aux limites de la transformation et d'améliorer la sélectivité en fréquence du banc de filtres. Les valeurs de la fonction fenêtre sont données dans le Tableau 47. Il est à noter que les 256 coefficients donnés sont utilisés dos à dos de manière à constituer une fenêtre symétrique de 512 points.

#### 8.2.3.2 Transformation temps-fréquence

Chaque bloc audio est transformé vers le domaine fréquence en effectuant une longue transformation avec  $N = 512$  points, ou deux courtes transformations avec  $N = 256$  points, en fonction de l'indicateur de commutation de bloc. Si  $x[n]$  représente la séquence d'entrée après fenêtrage, la séquence de fréquences en sortie  $X_D[k]$  est donnée par:

$$X_D[k] = \frac{-2}{N} \sum_{n=0}^{N-1} x[n] \cos \left( \frac{2\pi}{4N} (2n+1)(2k+1) + \frac{\pi}{4} (2k+1)(1+\alpha) \right) \quad \text{pour } 0 \leq k < N/2$$

où:

$\alpha = -1$  pour la première courte transformation

0 pour la longue transformation

+1 pour la seconde courte transformation.

## 8.2.4 Stratégie de couplage

### 8.2.4.1 Codeur élémentaire

Un codeur élémentaire peut utiliser une stratégie de couplage statique. Des paramètres de couplage convenables sont:

```

cplbegf    = 6; /* le couplage commence à 10,2 kHz */
cplendf    = 12; /* le canal de couplage se termine à 20,3 kHz */
cplbndstrc = 0, 0, 1, 1, 0, 1, 1, 1;
cplinu     = 1; /* le couplage est toujours actif */
/* tous les canaux sans commutation de bloc sont couplés */
for(ch=0; ch<nfchans; ch++) if(blksw[ch]) chincpl[ch] = 0; else chincpl[ch] = 1.

```

Les coordonnées de couplage peuvent être transmises un bloc sur deux, c'est-à-dire pour les blocs 0, 2 et 4. Les blocs 1, 3, et 5 réutilisent les coordonnées de couplage.

### 8.2.4.2 Codeurs évolués

Des codeurs plus évolués peuvent utiliser des paramètres de couplage dynamiques. Les fréquences de couplage peuvent varier en fonction d'une exigence de bits et d'un modèle psychoacoustique qui comparent l'audibilité des défauts créés par les techniques de réduction binaire par rapport à ceux créés par le processus de couplage. Les canaux dont le niveau de puissance varie rapidement avec le temps peuvent être sortis du couplage. Les coordonnées de couplage des canaux dont les niveaux de puissance varient lentement peuvent être envoyées moins souvent. La structure de la bande de couplage peut être rendue dynamique.

## 8.2.5 Constitution du canal de couplage

### 8.2.5.1 Canal de couplage

Le codeur le plus simple peut créer le canal de couplage en additionnant simplement les coefficients individuels de tous les canaux et en divisant le résultat par 8. La division par 8 évite qu'un coefficient du canal de couplage soit supérieur à 1. Des décodeurs un peu plus perfectionnés peuvent modifier les signes des divers canaux avant de les sommer afin d'éviter des annulations de phase.

### 8.2.5.2 Coordonnées de couplage

Les coordonnées de couplage sont créées en calculant des rapports de puissance dans chaque bande de couplage. La puissance dans le canal d'origine au sein d'une bande de couplage est divisée par la puissance dans le canal de couplage au sein de la bande de couplage. Ce rapport de puissance devient la coordonnée de couplage. Les coordonnées de couplage sont converties en format de virgule fixe et quantifiées. Les exposants de chaque canal sont examinés pour déterminer s'ils peuvent être mis à l'échelle par un facteur de 3, 6 ou 9. Ceci génère une coordonnée directrice de couplage pour le canal. (La coordonnée directrice de couplage permet d'augmenter la plage dynamique de la coordonnée de couplage.)

### 8.2.6 Rematriçage

Le rematriçage n'est actif que dans le mode 2/0. Des mesures de puissance des signaux L, R, L+R et L-R sont faites au sein de chaque bande de rematriçage. L'indicateur de rematriçage n'est pas positionné si la puissance maximale se trouve dans les canaux L ou R. L'indicateur de rematriçage est positionné si la puissance maximale se trouve dans les canaux L+R ou L-R. Lorsque l'indicateur de rematriçage est positionné, le codeur code L+R et L-R au lieu de L et R. Le rematriçage est décrit au § 7.5.

### 8.2.7 Extraction des exposants

La représentation binaire de chaque coefficient de fréquence est examinée en vue de déterminer le nombre de zéros en tête. Le nombre de zéros en tête (jusqu'à 24) donne la valeur de l'exposant initial. Ces exposants sont extraits et les ensembles d'exposants (un pour chaque bloc de chaque canal, y compris le canal de couplage) sont utilisés pour déterminer les stratégies d'exposant appropriées.

### 8.2.8 Stratégie d'exposant

Pour chaque canal, la variation des exposants en fonction de la fréquence et du temps est analysée. Si pour les exposants le spectre est relativement plat, une stratégie d'exposant telle que D25 ou D45 peut être utilisée. Si le spectre est très tonal, alors une stratégie d'exposant à haute résolution spectrale telle que D15 ou D25 devrait être utilisée. Si le spectre change peu pendant les 6 blocs d'une trame, les exposants peuvent n'être transmis que pour le bloc 0 et réutilisés pour les blocs 1-5. Si les exposants varient rapidement au cours d'une trame, les exposants peuvent être transmis pour le bloc 0 et pour les blocs qui ont des ensembles d'exposants très différents de ceux envoyés précédemment. Il existe un compromis entre une résolution fine en fréquence, une résolution temporelle fine et le nombre de bits requis pour transmettre les exposants. Lorsqu'on fonctionne à des débits binaires très faibles, il est en général nécessaire d'établir un compromis entre la résolution temporelle et la résolution en fréquence.

Un codeur élémentaire peut utiliser un algorithme simple. Il faut examiner tout d'abord la variation des exposants dans le temps. De nouveaux exposants sont envoyés si la variation excède un certain seuil. La stratégie d'exposant utilisée dépend du nombre de blocs pour lesquels le nouvel ensemble d'exposants est employé. Utiliser la stratégie D45 si les exposants sont employés pour un bloc uniquement. Utiliser la stratégie D25 si les exposants sont employés pour 2 ou 3 blocs. Utiliser la stratégie D15 si les nouveaux exposants sont employés pour 4, 5 ou 6 blocs.

### 8.2.9 Stratégie de bruit de désambiguïsation (dither)

Pour chaque canal, la décision de reproduire par un bruit de désambiguïsation (dither) les coefficients dont les valeurs quantifiées sont nulles est prise au niveau du codeur. Le but visé est de maintenir approximativement la même énergie dans le spectre reproduit, même s'il n'y a pas de bits alloués pour des portions du spectre. Il peut être intéressant de ne pas utiliser le bruit de désambiguïsation (dither) pour certains blocs, compte tenu de la stratégie d'exposants et de la précision des exposants codés.

Un codeur élémentaire peut mettre en œuvre une stratégie simple de bruit de désambiguïsation (dither) au niveau de chaque canal. Lorsque `blksw[ch]` est égal à 1, le bruit de désambiguïsation (dither) pour ce bloc et le suivant est inactivé.

### 8.2.10 Codage des exposants

Les exposants de chaque ensemble d'exposants sont prétraités en fonction de la stratégie d'exposant choisie. Les stratégies d'exposant D25 et D45 imposent qu'un exposant unique soit commun à plusieurs mantisses. Les exposants sont codés d'une manière différentielle en vue de leur transmission dans le flux binaire. La différence entre deux exposants absolus successifs ne produit pas nécessairement un code de différence valide (de valeur maximale  $\pm 2$ ) si le taux de variation des exposants absolus est supérieur à celui autorisé par la stratégie d'exposant. Le prétraitement ajuste les exposants de manière à ce que les coefficients de transformée qui ont en commun un exposant aient les mêmes exposants et que les différences aient des valeurs valides. Ceci a pour effet de diminuer la valeur de certains exposants et d'introduire des zéros en tête des mantisses correspondantes.

Les exposants subissent un codage différentiel générant l'enveloppe spectrale codée. Une partie du traitement du codeur produit un ensemble d'exposants identique à l'ensemble d'exposants dont dispose le décodeur lorsqu'il décode l'enveloppe spectrale codée.

### 8.2.11 Mantisses normalisées

Les coefficients de la transformée de chaque canal sont normalisés en décalant vers la gauche chaque coefficient du nombre de positions indiquées par l'exposant correspondant pour donner les mantisses normalisées. Les coefficients de fréquence sous leur forme binaire d'origine sont décalés vers la gauche en fonction des exposants qui seront utilisés par le décodeur. Quelques-unes des mantisses normalisées contiennent des zéros en tête. Les mantisses normalisées sont les données qui subissent la quantification.

### 8.2.12 Affectation centrale de bits

Un codeur élémentaire peut utiliser une procédure d'affectation centrale de bits dont tous les paramètres ont des valeurs nominales fixées par défaut:

```

sdcycod = 2;
fdccod = 1;
sgaincod = 1;
dbpbcod = 2;
floorcod = 4;
cplfgaincod = 4;
fgaincod[ch] = 4;
lfegaincod = 4;
cplsnroffst = fsnroffst[ch] = lfsnroffst = fineoffset;
```

Comme tous les paramètres d'affectation de bits sont statiques, ils ne sont envoyés que pour le bloc 0. L'affectation de bits delta n'est pas utilisée, de sorte que `deltbaie = 0`. Le sous-programme central d'affectation de bits (décrit dans le § 7.2) est exécuté et les déplacements de rapport S/N globaux et fins sont ajustés jusqu'à ce que tous les bits de la trame soient utilisés. Le déplacement de rapport S/N global est ajusté par incréments de 6 dB et le déplacement fin est ajusté par incréments de 3/8 de dB. Les bits sont attribués d'une manière globale pour tous les canaux à partir d'un ensemble commun de bits. Les combinaisons de déplacements `csnroffst` et `fineoffset` sont choisies de manière à utiliser le plus grand nombre de bits sans excéder la taille de la trame. Cette opération fait intervenir un processus itératif. Lorsque le nombre de bits est supérieur à la taille de l'ensemble commun lors d'une itération donnée, le déplacement S/N est diminué lors de l'itération suivante. D'une manière symétrique, si l'affectation est inférieure de l'ensemble commun,

le déplacement S/N est augmenté pour l'itération suivante. Lorsque le déplacement S/N atteint sa valeur maximale sans que l'affectation dépasse la taille de l'ensemble commun, l'itération est terminée. Les résultats du programme d'affectation de bits sont les valeurs de déplacements `csnroffst` et `fineoffset` ainsi que l'ensemble des pointeurs d'affectation de bits (`baps`). Les valeurs du déplacement S/N sont incluses dans le flux binaire de sorte que le décodeur n'a pas à faire d'itération.

### 8.2.13 Quantification des mantisses

Les `bap` sont utilisés par le bloc de quantification de mantisses. Il existe un `bap` pour chaque coefficient de la transformée. Chaque mantisse normalisée est quantifiée par le quantifieur indiqué dans le `bap` correspondant. Les matrices quantifiées d'une manière asymétrique sont quantifiées en arrondissant le nombre de bits indiqués dans le `bap` correspondant. Les matrices quantifiées d'une manière symétrique sont quantifiées en utilisant une consultation de table. Les mantisses avec des `bap` de 1, 2 et 4 sont regroupées en triplets ou en paires.

### 8.2.14 Empaquetage de la trame AC-3

Toutes les données sont empaquetées dans la trame AC-3 codée. Certaines des mantisses quantifiées sont regroupées et transmises dans un mot de code unique. Le format de sortie dépend de l'application. La trame peut être envoyée en rafale ou transmise sous la forme de données série avec un débit constant.

## APPENDICE 1 DE L'ANNEXE 2

(Normative)

### Flux élémentaires AC-3 dans un multiplex MPEG-2

## 1 Domaine d'application

Le présent Appendice contient des spécifications concernant la façon de combiner un ou plusieurs flux élémentaires AC-3 dans un «flux de transport» ou un «flux de programme» MPEG-2 (ISO/CEI 13818-1). Pour les applications qui citent cette spécification, il pourrait être nécessaire de préciser les valeurs de certains des paramètres décrits dans le présent Appendice.

## 2 Introduction

Le flux binaire élémentaire AC-3 est inclus dans le flux binaire d'un multiplex MPEG-2 de la même manière qu'un flux audio MPEG-1. Le flux binaire AC-3 est mis en paquets PES. Le flux binaire d'un multiplex MPEG-2 contenant des flux élémentaires AC-3 doit satisfaire à toutes les contraintes audio décrites dans le modèle STD dans le § 3.6. Il est nécessaire d'indiquer sans ambiguïté qu'un flux AC-3 est, en fait, un flux AC-3 (et non un flux audio MPEG). La norme MPEG-2 n'indique pas explicitement les codes à utiliser pour signaler un flux AC-3. De même, la norme MPEG-2 ne comprend pas de descripteur audio capable de décrire le contenu du flux binaire AC-3 dans les tableaux PSI.

L'unité d'accès (UA) ou l'unité de présentation (UP) audio AC-3 est une trame de synchronisation AC-3. Celle-ci contient 1 536 échantillons audio. La durée d'une unité d'accès (ou de présentation) AC-3 est de 32 ms pour un signal audio échantillonné à 48 kHz, de 34,83 ms environ pour un signal audio échantillonné à 44,1 kHz et de 48 ms pour un signal audio échantillonné à 32 kHz.

Les éléments qui doivent être spécifiés afin d'inclure un signal AC-3 dans le flux binaire MPEG-2 sont: `stream_type`, `stream_id`, le descripteur d'enregistrement, et le descripteur audio AC-3. L'utilisation du descripteur de langage ISO 639 est facultative. La couche PES est soumise à quelques contraintes dans le cas des flux audio multiples destinés à être reproduits en synchronisant exactement les échantillons.



### 3 Spécification détaillée

#### 3.1 Stream\_type

La valeur préférée de l'élément `stream_type` pour un signal AC-3 est 0x81. D'autres valeurs que le MPEG a qualifiées de «réservée aux utilisateurs» peuvent également être utilisées. Si la valeur 0x81 est utilisée, le décodeur peut supposer, en fonction de l'application, que l'élément `stream_type` 0x81 indique un signal audio AC-3. S'il y a un risque d'ambiguïté, le descripteur d'enregistrement AC-3 devrait être inclus (voir le § 3.3).

#### 3.2 Stream\_id

##### 3.2.1 Flux de transport

Pour les flux de transport, la valeur de l'élément `stream_id` dans l'en-tête PES sera 0xBD (indiquant l'élément `private_stream_1`). Les flux multiples AC-3 peuvent présenter la même valeur en ce qui concerne `stream_id` puisque chaque flux transporté est assorti d'une valeur PID unique. Les valeurs PID et l'élément `stream_type` sont mis en correspondance dans la table de correspondances du programme de flux de transport (PMT).

##### 3.2.2 Flux de programmes

Dans les flux de programmes, l'élément `stream_type` est destiné à spécifier le type et le numéro du flux élémentaire. Les flux élémentaires multiples AC-3 ne peuvent pas utiliser la même valeur en ce qui concerne `stream_id`; des valeurs uniques sont nécessaires. Si un seul flux élémentaire AC-3 est transporté dans un flux de programmes, l'élément `stream_id` peut utiliser la valeur 0xBD (indiquant l'élément `private_stream_1`). La norme ISO/CEI 13818-1 ne fournit pas de valeurs de `stream_id` permettant d'identifier les flux élémentaires multiples AC-3. Si de tels flux sont transportés dans un flux de programme l'élément `stream_id` utilisera les valeurs 110x xxxx, où x xxxx indique un numéro de flux compris entre 0 et 31. Cette valeur de l'élément `stream_id` est identique à celle qui est utilisée pour un signal audio MPEG-1 ou MPEG-2. Pour éviter toute confusion entre un signal audio MPEG et un signal audio AC-3, on peut utiliser une table de correspondances du flux de programmes, qui associe les valeurs de l'élément `stream_id` à celles de `stream_type`. Les flux qui utilisent un élément `stream_id` de 110x xxxx sont clairement identifiés quant au type de codage audio employé par la valeur de `stream_type` qui est liée à chaque valeur de `stream_id`.

#### 3.3 Descripteur d'enregistrement

La syntaxe du descripteur d'enregistrement AC-3 est présentée dans le Tableau 49. Si la valeur de l'élément `stream_type` qui est utilisée pour le signal AC-3 n'est pas 0x81, le descripteur d'enregistrement AC-3 sera inclus dans `TS_program_map_section` (pour les flux de transport) ou dans `program_stream_map` (pour les flux de programmes). Si la valeur `stream_type` utilisée pour le signal AC-3 est 0x81, le descripteur d'enregistrement AC-3 peut être inclus (il devrait l'être s'il existe un quelconque risque d'ambiguïté).

TABLEAU 49

Descripteur d'enregistrement AC-3

Syntaxe	Nombre de bits	Mnémonique
<pre>registration_descriptor() {   descriptor_tag   descriptor_length   format_identifier }</pre>	<p>8</p> <p>8</p> <p>32</p>	<p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p>

`descriptor_tag` — 0x05.

`descriptor_length` — 0x04.

`format_identifier` — Le `format_identifier` AC-3 est 0x41432D33 («AC-3»).

#### 3.4 Descripteur audio AC-3

Le descripteur audio AC-3, présenté dans le Tableau 50, permet d'inclure des informations concernant les différents flux élémentaires AC-3 dans les tableaux d'informations spécifiques aux programmes (PSI). Ces informations sont utiles pour orienter le(s) flux AC-3 approprié(s) vers le décodeur audio. Il convient de noter que les lignes horizontales du Tableau indiquent les points de terminaison admissible pour le descripteur.

TABLEAU 50

## Syntaxe de descripteur audio AC-3

Syntaxe	Nombre de bits	Mnémonique
audio_stream_descriptor() { descriptor_tag descriptor_length sample_rate_code bsid bit_rate_code surround_mode bsmod num_channels full_svc	8 8 3 5 6 2 3 4 1	uimbsf uimbsf bslbf bslbf bslbf bslbf bslbf bslbf bslbf
langcod	8	bslbf
if(num_channels==0) /* 1+1 mode */ langcod2	8	bslbf
if(bsmod<2) { mainid reserved } else asvcflags	3 5 8	uimbsf bslbf bslbf
textlen text_code for(i=0; i<M; i++) { text[i] }	7 1 8	uimbsf bslbf bslbf
for(i=0; i<N; i++) { additional_info[i] } }	N×8	bslbf

**descriptor\_tag** – La valeur préférée pour l'étiquette du descripteur AC-3 est 0x81. D'autres valeurs qualifiées par le MPEG comme étant réservées aux utilisateurs peuvent également être utilisées.

**descriptor\_length** – Il s'agit d'un champ à 8 bits qui indique le nombre de bits du descripteur suivant immédiatement le champ **descriptor\_length**.

**sample\_rate\_code** – Il s'agit d'un champ à 3 bits qui indique la fréquence d'échantillonnage du signal audio codé. Il peut s'agir d'une fréquence d'échantillonnage spécifique ou d'un ensemble de valeurs comprenant la fréquence d'échantillonnage du signal audio codé. Voir le Tableau 51.

TABLEAU 51

## Tableau des codes des fréquences d'échantillonnage

sample_rate_code	Fréquence d'échantillonnage (kHz)
'000'	48
'001'	44,1
'010'	32
'011'	Réservé
'100'	48 ou 44,1
'101'	48 ou 32
'110'	44,1 ou 32
'111'	48 ou 44,1 ou 32

**bsid** – Il s'agit d'un champ à 5 bits mis à la même valeur que le champ **bsid** dans le flux élémentaire AC-3.

**bit\_rate\_code** – Il s'agit d'un champ à 6 bits. Les cinq bits les moins significatifs indiquent un débit binaire nominal. Le bit le plus significatif (MSB) révèle si le débit binaire indiqué est exact (MSB = 0) ou indique une limite supérieure (MSB = 1). Voir le Tableau 52.

TABLEAU 52

Tableau des codes des débits binaires

bit_rate_code	Débit binaire exact (kbit/s)	bit_rate_code	Débit binaire maximal (kbit/s)
'000000' (0.)	32	'100000' (32.)	32
'000001' (1.)	40	'100001' (33.)	40
'000010' (2.)	48	'100010' (34.)	48
'000011' (3.)	56	'100011' (35.)	56
'000100' (4.)	64	'100100' (36.)	64
'000101' (5.)	80	'100101' (37.)	80
'000110' (6.)	96	'100110' (38.)	96
'000111' (7.)	112	'100111' (39.)	112
'001000' (8.)	128	'101000' (40.)	128
'001001' (9.)	160	'101001' (41.)	160
'001010' (10.)	192	'101010' (42.)	192
'001011' (11.)	224	'101011' (43.)	224
'001100' (12.)	256	'101100' (44.)	256
'001101' (13.)	320	'101101' (45.)	320
'001110' (14.)	384	'101110' (46.)	384
'001111' (15.)	448	'101111' (47.)	448
'010000' (16.)	512	'110000' (48.)	512
'010001' (17.)	576	'110001' (49.)	576
'010010' (18.)	640	'110010' (50.)	640

**dsurmod** – Il s'agit d'un champ à 2 bits qui peut être mis à la même valeur que le champ **dsurmod** dans le flux élémentaire AC-3 ou à '00' (pas d'indication). Voir le Tableau 53.

TABLEAU 53

Tableau dsurmod

surround_mode	Sens
'00'	Pas d'indication
'01'	PAS de codage ambiophonique Dolby
'10'	Codage ambiophonique Dolby
'11'	Réservé

**bsmod** – Il s'agit d'un champ à 3 bits qui est mis à la même valeur que le champ **bsmod** dans le flux élémentaire AC-3.

**num\_channels** – Il s'agit d'un champ à 4 bits qui indique le nombre de canaux dans le flux élémentaire AC-3. Lorsque le MSB est mis à 0, les 3 bits les moins significatifs sont mis à la même valeur que le champ **acmod** dans le flux élémentaire AC-3. Lorsque le champ du MSB est mis à 1, les 3 bits les moins significatifs indiquent le nombre maximal de canaux audio codés (en comptant le canal lfe comme 1). Si la valeur du champ **acmod** dans le flux élémentaire AC-3 est '000' (mode 1 + 1), la valeur du champ **num\_channels** sera mis à '0000'. Voir le Tableau 54.

TABLEAU 54

Tableau num\_channels

num_channels	Mode de codage audio (acmod)	num_channels	Nombre de canaux codés
'0000'	1 + 1	'1000'	1
'0001'	1/0	'1001'	≤ 2
'0010'	2/0	'1010'	≤ 3
'0011'	3/0	'1011'	≤ 4
'0100'	2/1	'1100'	≤ 5
'0101'	3/1	'1101'	≤ 6
'0110'	2/2	'1110'	Réservé
'0111'	3/2	'1111'	Réservé

**full\_svc** – Il s'agit d'un champ à 1 bit qui indique si ce service audio est un service complet susceptible d'être présenté ou s'il ne s'agit que d'un service partiel qui devrait être combiné avec un autre service audio avant d'être présenté. Ce bit devrait être mis à '1' si ce service audio est suffisamment complet pour être présenté à l'auditeur sans être combiné avec un autre service audio (par exemple, un service pour malvoyants contenant tous les éléments du programme; musique, effets, dialogue, et la description vocale du contenu visuel). Ce bit devrait être mis à '0' si le service, n'étant pas suffisamment complet, doit être présenté, combiné avec un autre service audio (par exemple, un service pour malvoyants qui ne contient qu'une description vocale du contenu visuel du programme et qui doit être combiné avec un autre service audio contenant la musique, les effets et le dialogue).

**langcod** – Il s'agit d'un champ à 8 bits qui est mis à la même valeur que le champ **langcod** dans le flux élémentaire AC-3. Une valeur de 0x00 indique que la langue n'est pas connue ou n'est pas indiquée.

**langcod2** – Il s'agit d'un champ à 8 bits qui est mis à la valeur du champ **langcod2** dans le flux élémentaire AC-3. Ce champ indique la langue du signal audio contenu dans le second canal audio monophonique (mode 1 + 1 seulement).

**mainid** – Il s'agit d'un champ à 3 bits qui contient un numéro compris entre 0 et 7 qui identifie un service audio principal. Chaque service principal devrait être étiqueté au moyen d'un numéro unique. Cette valeur est utilisée comme identificateur pour lier les services associés aux principaux services.

**asvclags** – Il s'agit d'un champ à 8 bits. Chaque bit (0-7) indique quel (quels) est (sont) le (les) service(s) principal (principaux) auquel (auxquels) est lié ce service associé. Le bit le plus à gauche (le bit 7) indique si ce service associé peut être reproduit avec le service principal N° 7. Si le bit a une valeur de 1, le service est associé avec le service principal N° 7. Si le bit a une valeur de 0, le service n'est pas associé avec le service principal N° 7.

**textlen** – Il s'agit d'un nombre entier non signé qui indique la longueur, exprimée en octets, du champ du texte descriptif qui suit.

**text\_code** – Il s'agit d'un champ à 1 bit qui indique comment le champ du texte suivant est codé. Si ce bit a une valeur '1', le texte est codé au moyen de caractères à 1 octet en utilisant l'alphabet latin-1 de l'ISO (ISO 8859-1). Si ce bit a une valeur '0', le texte est codé au moyen de caractères unicodes à 2 octets.

**text[j]** – Le champ de texte peut contenir un bref texte descriptif du service audio.

**additional\_info[j]** – Il s'agit d'un ensemble d'octets additionnels qui remplissent le reste du descripteur. Le rôle de ces octets n'est pas encore défini. Ce champ est prévu en vue de pouvoir élargir le descripteur dans l'avenir.

### 3.5 ISO\_639\_language\_code

Le descripteur ISO\_639\_language\_code permet d'étiqueter un flux au moyen du code de langue ISO 639 à 24 bits. Le flux binaire AC-3 et le descripteur audio AC-3 contiennent tous deux un code de langue (identique) à 8 bits qui se prête à la plupart des applications. Il est donc redondant d'utiliser en plus le descripteur ISO\_639\_language\_code. Si ISO\_639\_language\_descriptor est inclus dans TS\_program\_map\_section (pour les flux de transport) ou dans program\_stream\_map (pour les flux de programme), le champ audio\_type de ce descripteur aura une valeur de 0x00 (non définie).

### 3.6 Taille de la mémoire tampon audio STD

Pour un flux de transport MPEG-2, le modèle T-STD définit la taille de la mémoire tampon audio principale,  $BS_n$ , comme suit:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh}$$

où:

$$BS_{mux} = 736 \text{ octets}$$

$BS_{oh}$  : surdébit de l'en-tête PES

$BS_{dec}$  : mémoire tampon de l'unité d'accès.

MPEG-2 spécifie une valeur fixe pour  $BS_n$  (3584 octets) et indique que toute mémoire tampon excédentaire peut être utilisée pour un multiplexage additionnel.

Lorsqu'un flux élémentaire AC-3 est transporté par un flux de transport MPEG-2, le flux de transport doit être conforme à la taille de la mémoire tampon audio principale suivante:

$$BS_n = BS_{mux} + BS_{pad} + BS_{dec}$$

où:

$$BS_{mux} = 736 \text{ octets}$$

$$BS_{pad} = 64 \text{ octets}$$

On peut trouver  $BS_{dec}$  dans le Tableau 13 de la Norme ATSC A/52 (dans le cas de la fréquence d'échantillonnage de 44,1 kHz, la plus élevée des deux valeurs proposées sera utilisée). Les 64 octets que compte  $BS_{pad}$  sont disponibles pour  $BS_{oh}$  et pour un multiplexage additionnel. Cette contrainte permet d'utiliser les décodeurs avec le minimum de mémoire tampon.

Les applications qui font appel aux flux de programmes devraient spécifier les contraintes en question.

## 4 Contraintes PES

### 4.1 Codage

Dans certaines applications, le décodeur audio peut décoder simultanément deux flux élémentaires contenant différents éléments de programme puis combiner ces éléments pour en faire un programme complet. La plupart des éléments de programme se trouvent dans le *service audio principal*. On peut trouver un autre élément de programme (tel que la description vocale du contenu de l'image à l'intention de l'auditeur malvoyant) dans le *service audio associé*. Dans ce cas, le décodeur audio peut séquentiellement décoder des trames audio (ou blocs audio) de chaque flux élémentaire et les combiner par trame (ou par bloc). Pour pouvoir reproduire le signal audio des deux flux élémentaires en synchronisant correctement les échantillons, il est nécessaire que les codeurs des flux élémentaires audio d'origine aient codé les deux trames d'élément de programme audio de manière synchrone; autrement dit, si le flux audio 1 comprend l'échantillon 0 de la trame  $n$  pris au moment  $t_0$ , le flux audio 2 devrait également comprendre la trame  $n$  à partir de son échantillon 0 pris au même moment  $t_0$ . Si le codage des services audio multiples est exécuté avec synchronisation des trames et des échantillons, et que le décodage est censé être effectué avec synchronisation des trames et des échantillons, les paquets PES de ces services audio comprendront des valeurs identiques de PTS renvoyant aux unités d'accès audio prévues pour le décodage synchrone.

Les services audio destinés à être combinés aux fins de reproduction seront codés en utilisant la même fréquence d'échantillonnage.

## 4.2 Décodage

Si les unités d'accès audio de deux services audio qui doivent être décodés simultanément possèdent des valeurs PTS identiques indiquées dans leurs en-têtes PES respectifs, les unités d'accès audio correspondantes seront présentées au décodeur audio en vue d'un décodage synchrone simultané. Un décodage synchrone signifie que, pour des trames (unités d'accès) audio, des échantillons audio correspondants sont présentés au même moment.

Si les valeurs PTS ne concordent pas (ce qui indiquerait que le codage audio n'a pas été effectué avec synchronisation des trames), les trames (unités d'accès) audio du service audio principal seront présentées au décodeur audio pour être décodées et présentées au moment indiqué par la valeur PTS. Un service associé décodé simultanément devrait avoir ses trames (unités d'accès) audio, qui sont le plus étroitement alignées dans le temps (comme indiqué par les valeurs PTS) sur celles du service principal qui est décodé, présentées au décodeur audio pour être décodées simultanément. Dans ce cas, le service associé peut être reproduit avec un décalage égal à la moitié de la durée de l'intervalle de trame. (Cela est généralement satisfaisant; une description vocale destinée aux malvoyants ne nécessite pas une synchronisation très précise.)

## 4.3 Alignement des bits

Le flux élémentaire AC-3 doit être aligné octet par octet avec le flux de données MPEG-2. Ceci signifie que les 8 premiers bits d'une trame AC-3 doivent être contenus dans un seul octet transporté par le flux de données MPEG-2.

---