

**TTC標準**  
Standard

JT-G719

高品質会話型用途向け  
低演算量フルバンドオーディオ符号化

Low-complexity full-band audio coding  
for high-quality conversational applications

第 1 版

2010 年 5 月 26 日制定

社団法人  
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE



本書は、(社) 情報通信技術委員会が著作権を保有しています。  
内容の一部又は全部を (社) 情報通信技術委員会の許諾を得ることなく複製、転載、  
改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

## 目 次

<参考> .....	5
1 本標準の規定範囲 .....	7
2 参考文献 .....	7
2.1 必須文献 .....	7
2.2 参考文献 .....	7
3 定義 .....	7
4 略語と頭字語 .....	7
5 表記法 .....	8
6 コーデックアルゴリズム概略 .....	10
6.1 符号化器 .....	10
6.2 復号器 .....	11
6.3 アルゴリズム遅延解析 .....	11
6.4 計算量、所要記憶容量 .....	12
6.5 コーデックの記述 .....	12
7 符号化器の機能記述 .....	13
7.1 過渡部検出 .....	13
7.2 適応時間-周波数変換 .....	14
7.2.1 入力信号バッファリング、および窓掛け .....	15
7.2.2 時間域エリアシング .....	16
7.2.3 信号事前正規化 .....	17
7.2.4 定常モード .....	18
7.2.5 過渡モード .....	18
7.2.6 高速離散コサイン変換タイプIVアルゴリズム .....	19
7.2.7 信号の事後正規化 .....	20
7.3 スペクトル係数のグルーピング .....	20
7.4 ノルム推定と量子化 .....	24
7.5 ビット割り当て .....	25
7.5.1 ビット割り当てのためのノルム事後調整 .....	25
7.5.2 ノルムに基づくビット割り当て .....	27
7.6 高速格子ベクトル量子化 .....	28
7.6.1 $REs$ に基づく格子ベクトル量子化 .....	28
7.6.2 $Ds$ に基づく格子ベクトル量子化 .....	30
7.7 雑音レベル調整 .....	33
7.8 ビット列形式 .....	35
8 復号器の機能説明 .....	35
8.1 フラグの復号 .....	35
8.2 ノルムの復号 .....	35
8.3 スペクトル係数の復号 .....	35
8.4 スペクトル充填 .....	36
8.4.1 雑音充填 .....	37
8.4.2 帯域拡張 .....	38
8.4.3 雑音レベル調整 .....	38
8.4.4 スペクトル充填包絡シェーピング .....	39
8.5 逆変換 .....	39
8.5.1 信号の事前正規化と事後正規化 .....	39
8.5.2 定常モード .....	40
8.5.3 過渡モード .....	40
8.5.4 窓処理と重ね合わせ加算 .....	41
8.6 フレーム消失補償 .....	41
9 柔軟なレート選択 .....	41
10 シミュレーションソフトウェア .....	42
10.1 シミュレーションソフトウェアの使用法 .....	42
10.2 シミュレーションソフトウェアの構成 .....	42
付属資料A 蓄積形式の定義 .....	46
A.1 適用範囲 .....	46
A.2 必須文献 .....	46
A.3 参考文献 .....	46
A.4 標準JT-G 719ビット列の蓄積形式 .....	46
A.4.1 単一チャンネル蓄積形式 .....	46

A. 4. 2	複数チャンネルオーディオのサポート	47
A. 5	ISOベースファイル形式のサンプルエントリ記述	48
A. 5. 1	サンプルエントリボックス	49
A. 5. 2	G719SampleEntryボックスに対するG719SpecificBox領域	49
付属資料B	浮動小数点実装	51
B. 1	適用範囲	51
B. 2	アルゴリズム記述	51
B. 3	ANSI Cコード	51
B. 3. 1	シミュレーションソフトウェアの使用方法	51
B. 3. 2	シミュレーションソフトウェアの構成	51
付属資料C	パケットフォーマット、能力識別子及び能力パラメータ	55
C. 1	参考文献	55
C. 2	標準JT-G719フレームのパケット構造	55
C. 3	標準JT-H245で使用する能力識別子及びパラメータ	55
C. 3. 1	標準JT-G719の能力識別子及びパラメータ	55
付録	用語対照表	58

## <参考>

### 0. 本標準の概要

本標準は、フルバンドの会話型音声・オーディオ信号に対し、32 kbit/s から 128 kbit/s までのビットレートで低演算量で動作する J T - G 7 1 9 符号化アルゴリズムを記述したものである。

符号化器入力信号および復号器出力信号が 48 kHz サンプリングで、20 Hz から 20 kHz までのフルバンドの、音声、音楽および音響信号を符号化することができる。

符号化アルゴリズムは、適応時間分解能、適応ビット割り当ておよび低演算量格子ベクトル量子化による変換符号化に基づくものである。

### 1. 国際勧告等との関連

本標準は、2008年6月に承認されたITU-T勧告G. 719に準拠したものである。また、本標準は、2008年11月および2009年3月に承認された、ITU-T勧告G. 719に対するAmendment 1およびAmendment 2に準拠したものである。

### 2. 上記国際勧告等に対する追加項目等

#### 2. 1 オプション選択項目

なし

#### 2. 2 ナショナルマター決定項目

なし

#### 2. 3 その他

- (1) 本標準は、上記ITU-T勧告に対し、先行している項目はない。
- (2) 本標準は、上記ITU-T勧告に対し、追加した項目はない。
- (3) 本標準は、上記ITU-T勧告に対し、削除した項目はない。
- (4) 本標準は、上記ITU-T勧告に対し、変更した項目はない。

#### 2. 4 原勧告との章立て構成比較

上記国際勧告等との章立て構成の相違はない。

### 3. 改版の履歴

版数	制定日	改版内容
第1版	2010年5月26日	制定

### 4. 工業所有権

本標準に関わる「工業所有権の実施の権利に係る確認書」の提出状況は、TTCホームページでご覧になれます。

### 5. その他

- (1) 参照している勧告、標準等

TTC標準： J T - G 7 2 2 . 1

ITU-T勧告： G . 1 9 1、G . 1 9 2

ISO/IEC: ISO/IEC 14496-12 | 15444-12

IETF: RFC3550、RFC5404

(2) TTC標準JT-G719は、ITU-T勧告G.719に準拠しており、本標準中で言及しているCコードおよびテストシーケンスとは、ITU-T勧告G.719のものをさし、ITU-TのWebサイトから入手可能である。

## 1 本標準の規定範囲

本標準は、フルバンドの音声／オーディオ信号に対する 32-128kbit/s での符号化アルゴリズムについて記載している。

本標準の構成は次の通りである。本標準を通して使用されている文献、略語、表記法については、第2章、第3章、第4章、第5章のそれぞれで定義している。第6章では JT-G 7 1 9 フルバンドアルゴリズムの概略について述べている。JT-G 7 1 9 フルバンド符号化／復号の原理については第7章、第8章のそれぞれで述べている。また、ビットレートおよび動作モードについては第9章で述べている。第10章では、16-32 ビット固定小数点演算で本符号化アルゴリズムを定義したソフトウェアについて記載している。

## 2 参考文献

### 2. 1 必須文献

下記の TTC 標準および ITU-T 勧告は、本標準での参照を通して本標準の規定を構成するものである。出版された時点でその版が適用され、全ての標準および他の参照文献は、改定に従うものとする。従って、本標準のユーザには、以下の TTC 標準やその他の参照すべき文献について、最新の版の適用の可能性を調査するよう奨励される。現在有効な TTC 標準および ITU-T 勧告のリストは定期的に出版されている。

本標準内での文書の参照は、単独の文書としては、それを標準の扱いとはしない。

- ITU-T 勧告 G. 1 9 1  
Software tools for speech and audio coding standardization
- ITU-T 勧告 G. 1 9 2  
A common digital parallel interface for speech standardization activities
- TTC 標準 JT-G 7 2 2. 1  
フレーム消失の少ないシステムにおけるハンズフリー用途向け 7kHz 帯域 24 および 32kbit/s オーディオ符号化方式

### 2. 2 参考文献

- [1] H. S. Malvar, "Signal Processing with Lapped Transforms", Norwood, MA: Artech House, 1992.
- [2] S. Winograd, "On computing the discrete Fourier transform", Math. Computation 32: 175-199, 1978.
- [3] J. H. Conway and N. J. A. Sloane, "Sphere Packings, Lattices and Groups", Springer-Verlag, New York, 1988.
- [4] C. Lamblin and J.-P. Adoul, "Algorithme de quantification vectorielle sphérique à partir du réseau de Gosset d'ordre 8", Annales des Télécommunications, 43, no 3-4, pp. 172-186, 1988.
- [5] D. A. Huffman, "A method for the construction of minimum-redundancy codes", Proceedings of IRE, vol. 40, pp. 1098-1101, September, 1952.

## 3 定義

本標準では特に用語の定義を行っていない。

## 4 略語と頭字語

本標準では、Table 1/JT-G719 で定義される略語と頭字語を使用している。

Table 1/JT-G719 – Glossary of acronyms  
(ITU-T G.719)

Acronym	Description
BWE	Bandwidth Extension
DCT	Discrete Cosine Transform
FB	Full Band
FLVQ	Fast Lattice Vector Quantization
IIR	Infinite Impulse Response
LVQ	Lattice Vector Quantization
MDCT	Modified Discrete Cosine Transform
MLT	Modulated Lapped Transform
MSE	Mean Squared Error
PDF	Probability Density Function
RAM	Random Access Memory
ROM	Read Only Memory
TDA	Time Domain Aliasing
WMOPS	Weighted Million Operations Per Second

## 5 表記法

Table 2/JT-G719.1 は、本標準で用いられている主な記号のリストである。

Table 2/JT-G719 – Glossary of most relevant symbols  
(ITU-T G.719)

Symbol	Description
$x(n)$	Input signal sampled at 48 kHz
$H_{HP}(z)$	High-pass filter used for the transient detection
$x_{HP}(n)$	Output of the high-pass filter $H_{HP}(z)$
$m$	Index of sub-frame
$E(m)$	Signal's short term energy corresponding to the $m$ th sub-frame
$E_{LT}(m)$	Signal's long term energy corresponding to the $m$ th sub-frame
$\alpha$	Forgetting factor for computing $E_{LT}(m)$
$\rho$	Energy ratio threshold
$x_{OLD}(n)$	Input signal from the previous 20ms frame
$h(n)$	Window function for transform
$x_w(n)$	Windowed signal with $h(n)$
$\tilde{\mathbf{x}}$	TDA output signal matrix
$\mathbf{x}_w$	Windowed signal matrix
$\mathbf{I}_L$	Identity matrices of order $L$
$\mathbf{J}_L$	Time reversal matrices of order $L$
$\tilde{x}(n)$	TDA output signal
$Q_{max}$	Normalization index for signal pre-scaling and post-scaling



Symbol	Description
$\tilde{\tilde{x}}(n)$	Scaled TDA signal
$y(k)$	Spectral coefficients of the input frame
$L$	Frame length in samples
$v(n)$	Re-ordered signal
$z(p)$	Complex sequence of TDA signal
$\tilde{z}(p)$	Pre-modulated complex sequence of TDA signal
$\tilde{Z}(q)$	Scaled inverse complex Fourier transform of $\tilde{z}(p)$
$\bar{y}(k)$	Spectrum corresponds to $\tilde{\tilde{x}}(n)$
$p$	Index of sub-vector
$L_p$	Length of the pth sub-vector
$s_p$	Starting index of the spectral coefficients forming the pth sub-vector
$e_p$	Ending index of the spectral coefficients forming the pth sub-vector
$N(p)$	Norm or spectrum energy of the pth sub-vector
$I_N(p)$	Index of the pth quantized norm
$\Delta_N(p)$	Differential index between $I_N(p+1)$ and $I_N(p)$
$Spe(p)$	Spectrum function
$BSpe(b)$	Remapped spectrum function
$B_{MAX}$	Maximum number of bands for spectral mapping bands, 20.
$H_b$	Linear mapping slope for spectrum remapping
$T_b$	Linear mapping offset for spectrum remapping
$J_b$	Set of indices for spectrum remapping
$A(b)$	Pseudo absolute threshold of hearing
$WSpe(p)$	Offset spectrum for norm adjustment
$\tilde{I}_N^q(n)$	Index of the pth adjusted quantized norm
$I_N^q(n)$	Index of the pth quantized norm
$G$	Generator matrix for the lattice D8
$\mathfrak{S}$	Index set of transform coefficients allocated zero bits
$f_t$	Transition frequency
$z^q(k)$	Reconstructed transform coefficients
$y^q(k)$	Decoded normalized transform coefficients
$\tilde{x}^q(n)$	Reconstructed TDA signal
$\tilde{\mathbf{x}}_m^{qw}$	Reconstructed TDA signal matrix for sub-frame $m$
$R(p)$	Bit-allocation vector
$\Omega$	Number of remainder bits
$\Psi$	Bit budget
$\mathbf{y}_{p,l}$	The lth 8-dimensional vector belonging to the pth sub-vector
$\beta$	Optimal scaling factor in LVQ1
$\beta(p)$	Optimal scaling vector in LVQ2 for the pth sub-vector

表記法の詳細は次の通りである。

- (1) 時間領域の信号は、そのシンボルと丸括弧で括られたサンプル番号で記述する(例  $s(n)$ )。変数 $n$ は、サンプル番号である。
- (2) 周波数領域に変換された信号は、対応する時間領域の信号を大文字に変えることにより記述する(例  $S(k)$ は $s(n)$ の変換)。変数 $k$ は、係数の番号である。
- (3) 丸括弧で括られた上付きの添字は、時間に依存する変数に用いる(例  $g^{(m)}$ )。変数 $m$ はその前後関係よりフレーム番号、あるいはサブフレーム番号に対応しており、変数 $n$ はサンプル番号に対応している。
- (4) パラメータの範囲は、角括弧で括られた値で記述する。この値は境界値を含む(例  $[0.6, 0.9]$ )。
- (5) 符号関数  $sgn(x)$  は、以下のように引数の極性を示す。

$$sgn(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

- (6) 整数演算子  $\lfloor x \rfloor$  は負の方向への丸めを示す(例  $\lfloor x \rfloor = \max\{n \in \mathbb{Z} \mid x \geq n\}$ )。
- (7) 使用される10進の浮動小数点値は、16ビット固定小数点ANSI Cでの実装に使用された値を丸めたものである。

## 6 コーデックアルゴリズム概略

J T-G 7 1 9フルバンドコーデック拡張は、低演算量な変換ベースのオーディオコーデックである。48kHz サンプリングで動作し、20-20000Hz に渡るオーディオ帯域を提供する。符号化器は、16-bit リニアPCMの入力信号を20msのフレーム毎に処理し、その処理遅延は40msとなる。符号化アルゴリズムは、適応時間分解能、適応ビット割り当て、低演算量格子ベクトル量子化を利用する変換符号化方法に基づいている。さらに、復号器は、信号適応雑音充填、帯域拡張のいずれかにより非符号化スペクトル要素を再現する。

### 6.1 符号化器

J T-G 7 1 9符号化器のブロック図をFigure 1/JT-G719に示す。48kHz サンプリングの入力信号は、まず過渡部検出器で処理される。過渡部の検出結果に基づき、高周波数分解能変換または低周波数分解能変換が入力信号フレームに適用される。定常フレームに対して行われる適応変換は、修正離散コサイン変換に基づく処理である。非定常フレームに対しては、高時間分解能変換が用いられる。この時、追加遅延はなく、演算量の増加も非常に少ない。また、非定常フレームにおける時間分解能は5msフレームに相当する。

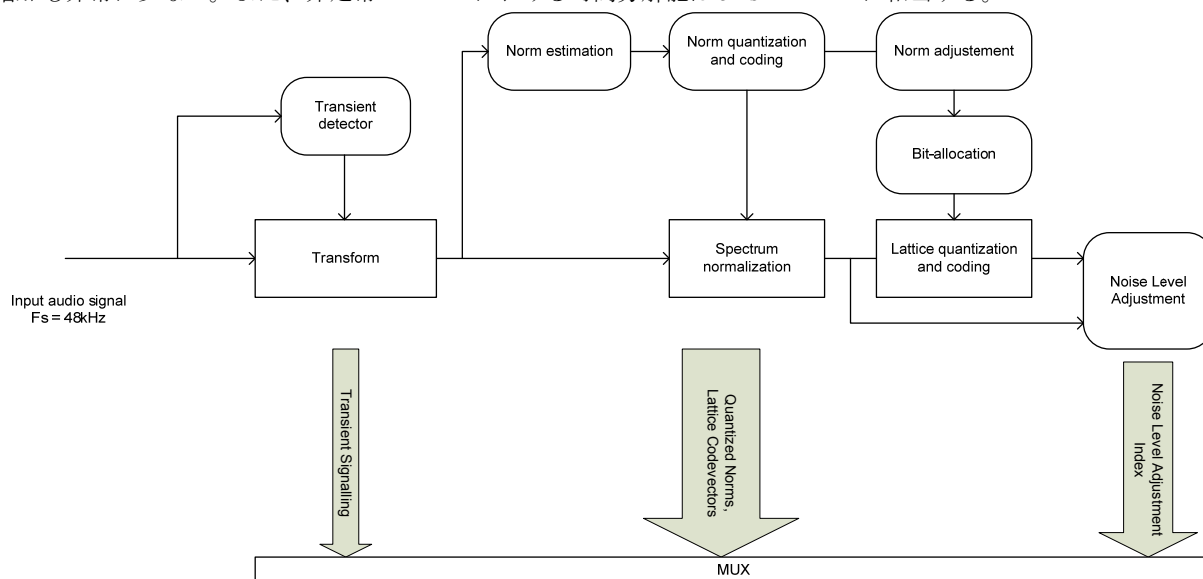


Figure 1/JT-G719-Encoder block diagram  
(ITU-T G.719)

上記により生成されるスペクトル係数は、非均等の複数帯域に分割される。各帯域のノルムが見積もられ、次に全帯域のノルムから構成されるスペクトル包絡が量子化され、符号化される。スペクトル係数は、量子化ノルムによって正規化される。また、量子化されたノルムはさらに適応スペクトル重みに基づいて調整され、ビット割り当て器に入力される。正規化されたスペクトル係数は格子ベクトル量子化され、各周波数帯域に対して割り当てられたビットに基づき符号化される。また、非符号化スペクトル係数のレベルの見積もりがなされる。これは量子化された後、復号器に伝送される。符号化されたノルムと同様に、上記の符号化スペクトル係数に対するインデックスの量子化には、ハフマン符号化が適用される。

## 6.2 復号器

J T-G 7 1 9 復号器のブロック図を Figure 2/JT-G719 に示す。まず過渡部フラグが復号される。ここで、過渡部フラグはフレーム構成（定常部/過渡部）を示す情報である。スペクトル包絡は復号され、また復号器において正規化されたスペクトル係数の量子化インデックスを復号するために不可欠なビット割り当てを再計算するために、ビットイグザクトであるノルム調整アルゴリズムおよびビット割り当てアルゴリズムが用いられる。逆量子化の後、受信したスペクトル係数（ビットが割り当てられたスペクトル係数）から構築されるスペクトル充填符号帳を用いて、低周波数の非符号化スペクトル係数（ビットが割り当てられなかったスペクトル係数）が再生成される。雑音レベル調整インデックスは再生成されたスペクトル係数のレベル調整に利用される。また、高周波数の非符号化スペクトル係数は帯域拡張により再生成される。復号スペクトル係数および再生成スペクトル係数は組み合わせられ、正規化スペクトルが算出される。次に、復号フルバンドスペクトルに対して復号スペクトル包絡が適用される。最後に、時間領域の信号を復元するために逆変換が適用される。これは、定常モードに対する修正離散コサイン逆変換か、あるいは過渡部モードに対する高時間分解能逆変換のいずれかの適用により実行される。

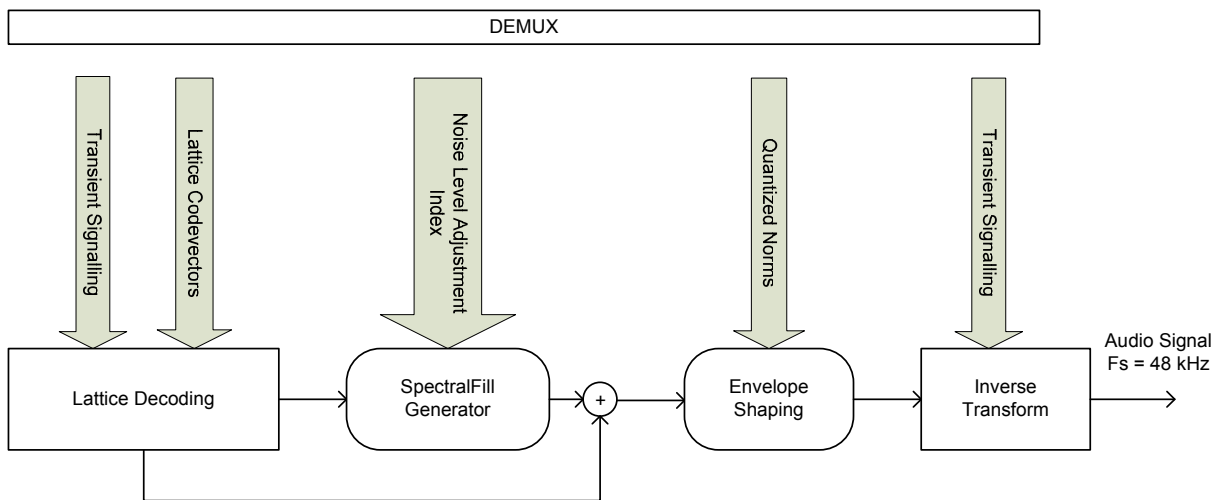


Figure 2/JT-G719 – Decoder block diagram  
(ITU-T G.719)

## 6.3 アルゴリズム遅延解析

J T-G 7 1 9 アルゴリズムは、適応変換符号化技術に基づくものである。この技術は、入出力オーディオ信号の 20ms フレーム毎に動作する。変換窓（基底関数長）は 40ms であり、連続する入出力フレーム間の 50% がオーバーラップしているため、効果的な先読みバッファのサイズは 20ms となる。それゆえ、全アルゴリズム遅

延は、フレームサイズと先読みサイズを加えた 40ms となる。また、処理遅延やネットワーク伝送遅延などが生じる場合に対しては、その他の付加的な遅延が発生する。

#### 6. 4 計算量、所要記憶容量

異なるビットレートに対して、符号化器および復号器で見積もられた J T - G 7 1 9 の平均計算量および最大計算量（単位は WMOPS）を Table 3/JT-G719 に示す。表中の数値は、I T U - T ソフトウェアツールライブラリ S T L 2 0 0 5 v 2 . 2 ( I T U - T 勧告 G . 1 9 1 ) の基本演算子を用いて算出された計算量報告に基づくものである。ここで前記報告は、J T - G 7 1 9 の主観品質特性試験において用いられたサンプルを符号化/復号した結果に基づいている。

Table 3/JT-G719 – Complexity of the G.719 codec  
(ITU-T G.719)

Bitrate (kbit/s)	Encoder only		Decoder only		Encoder plus Decoder	
	Average	Maximum	Average	Maximum	Average	Maximum
32	6.663	7.996	6.876	7.413	13.539	15.397
48	7.427	9.073	7.230	7.806	14.657	16.861
64	7.912	9.899	7.554	8.161	15.466	18.060
80	8.303	10.564	7.865	8.496	16.169	19.026
96	8.555	10.796	8.122	8.757	16.677	19.536
112	8.787	10.881	8.368	9.009	17.155	19.890
128	8.980	11.775	8.610	9.225	17.590	21.000

Table 4/JT-G719 は、J T - G 7 1 9 の 16-bit kwords でのメモリ容量(Static RAM, Scratch RAM, Data ROM)、および S T L ツールにより基本演算子の回数で測定されたプログラム ROM の値を示している。これらは、J T - G 7 1 9 の固定小数点 A N S I C ソースコードに基づいている。

Table 4/JT-G719 – Memory usage of the G.719 codec  
(ITU-T G.719)

Memory type	Encoder	Decoder	Codec (encoder plus decoder)
Static RAM (kWords)	1.0	3.9	4.9
Scratch RAM (kWords)	12.2	12.2	24.4
Data ROM (kWords)	8.3	8.9	10.7
Program ROM (1000's basic ops)	1.2	1.2	1.8

#### 6. 5 コーデックの記述

本標準の符号化アルゴリズムは、ビットイグザクトな固定小数点算術演算で記述されている。第 1 0 章で示される A N S I C コードは、本標準の必須部分を構成するものであり、このビットイグザクトな固定小数点での記述を反映している。コーデックの算術的な記述は、他の方法でも実装し得るが、本標準に準拠しないコーデックを実装することになってしまう可能性がある。

したがって、不一致が生じた場合には、算術的な記述よりも、第 1 0 章の A N S I C コードによるアルゴリズム記述の方が優先される。A N S I C コードと共に用いられるテスト信号のセットは、I T U - T の Web

サイトから入手可能である。

## 7 符号化器の機能記述

### 7.1 過渡部検出

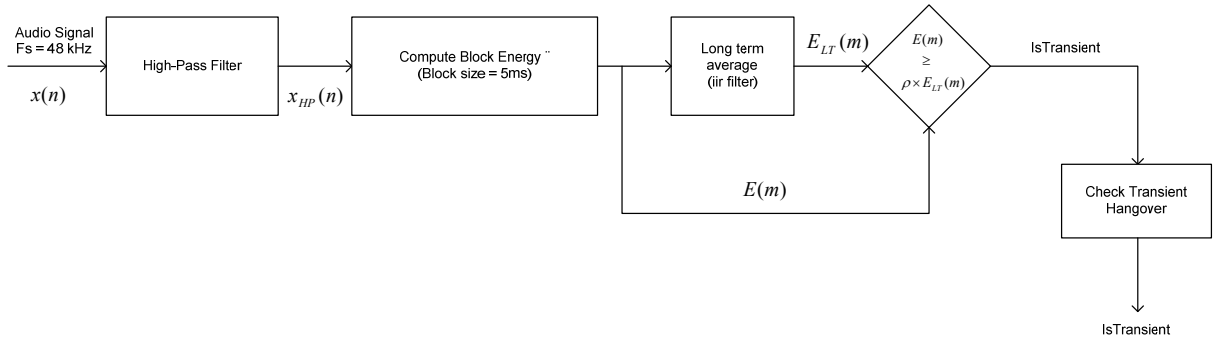


Figure 3/JT-G719 – Transient detection algorithm  
(ITU-T G.719)

J T – G 7 1 9 符号化器の過渡部検出アルゴリズムを Figure 3/JT-G719 に示す。48kHz サンプリングの入力信号  $x(n)$  は、まずハイパスフィルタ処理される。ここで、ハイパスフィルタは予期しない低周波数成分に対する予防策として用いられる。ハイパスフィルタとしては、以下に示す一次の I I R フィルタが用いられる。

$$H_{HP}(z) = \frac{0.7466(1 - z^{-1})}{1 - 0.4931z^{-1}} \quad (1)$$

上記ハイパスフィルタの周波数応答を Figure 4/JT-G719 に示す。

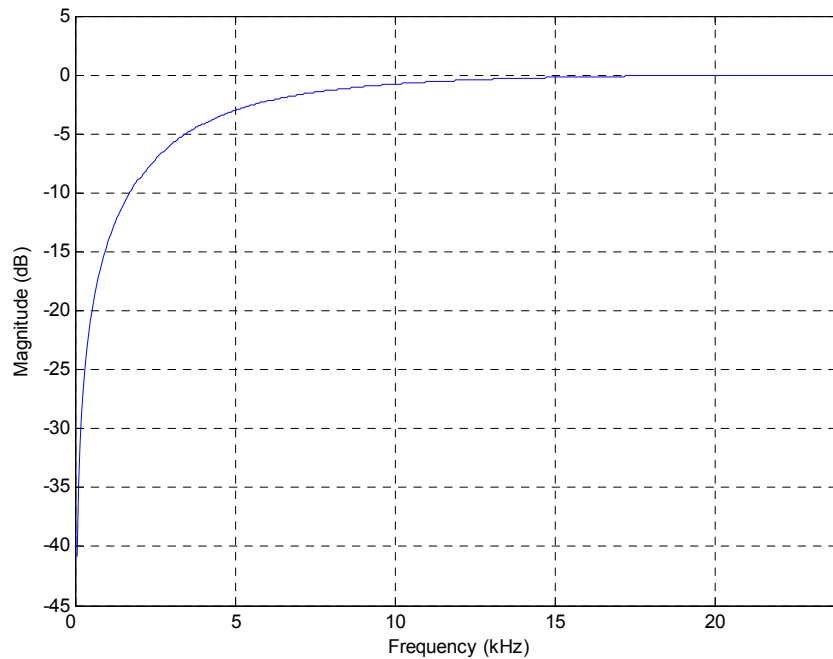


Figure 4/JT-G719 – Frequency response of high-pass filter  
(ITU-T G.719)

上記ハイパスフィルタの出力  $x_{HP}(n)$  は以下のようにして算出される。

$$x_{HP}(n) = 0.4931 x_{HP}(n-1) + 0.7466 x(n) - 0.7466 x(n-1), n = 0, \dots, L-1 \quad (2)$$

ここで、 $L=960$  は 20ms フレーム長であることを示す。ハイパスフィルタ処理された信号  $x_{HP}(n)$  は 5ms、つまり  $L/4=240$  サンプル毎のサブフレームに分割される。

次に、各サブフレームのエネルギー  $E(m)$  が以下のようにして算出される。

$$E(m) = \frac{1}{L/4} \sum_{n=mL/4}^{(m+1)L/4-1} x_{HP}(n)^2, m = 0, \dots, 3 \quad (3)$$

各サブフレームの信号に対する長期エネルギー  $E_{LT}(m)$  が以下の式により更新される。

$$E_{LT}(m) = (1 - \alpha) E_{LT}(m-1) + \alpha E(m), m = 0, \dots, 3 \quad (4)$$

上式において、 $\alpha$  は忘却係数であり 0.25 に設定され、また前フレームの値から  $E_{LT}(-1) = E_{LT}(3)$  のように更新される。また、次フレームにおける処理のために、ハイパスフィルタ  $H_{HP}(z)$  の状態および  $E_{LT}(3)$  が保存される。

各サブフレーム  $m$  に対して、短期エネルギー  $E(m)$  と長期エネルギー  $E_{LT}(m)$  の比較がなされる。そのエネルギー比がある閾値を超えた場合に、過渡部が検出される。過渡部は以下を満たす場合に検出される。

$$E(m) \geq \rho \times E_{LT}(m) \quad (5)$$

ここで、 $\rho$  はエネルギー比の閾値であり、 $10 \log_{10}(\rho) = 7.8 \text{ dB}$  に設定される。

一般的には、時間一周波数変換は 40ms フレームに対して適用される。それゆえ、過渡部は連続する 2 フレームに影響を与えてしまう。これを解決するために、過渡部検出にはハングオーバーが用いられている。あるフレームで検出された過渡部は、次フレームでの過渡部にも繋がる。

過渡部検出器の出力は *IsTransient* というフラグである。このフラグは、過渡部が検出された場合には論理値 *TRUE* に、そうでない場合は *FALSE* に設定される。

## 7. 2 適応時間-周波数変換

適応時間-周波数変換は、過渡性の判定結果に基づく。定常的な信号の場合は、定常的な音を効率的に表現できるように高い周波数解像度で変換が行われる。過渡的な音の場合は、時間解像度を高めた変換が行われ、それによって入力信号の周波数特性の急激な変化をより良く表現することができる。

これら 2 つのモードの処理は、共通のバッファリングモジュールと窓掛けモジュールを共有し、一方のモードから他方のモードへの切り替えは瞬時的に行われる。それゆえ、過渡性判定のための追加の先読みは必要ない。これによって、コーデックは低負荷かつ追加の遅延無しで選択的な時間解像度を持つことができる。

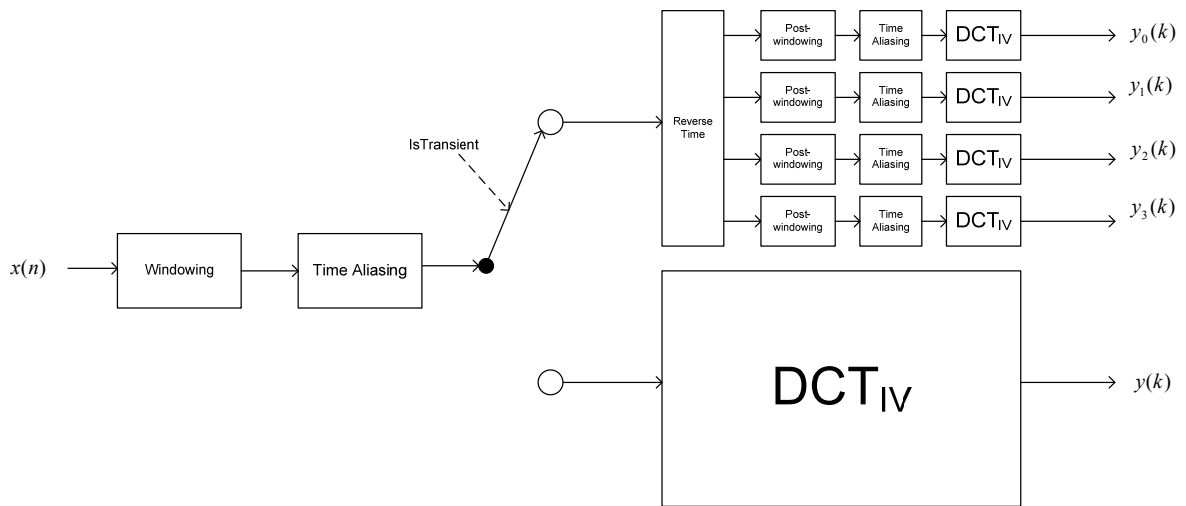


Figure 5/JT-G719 – Adaptive Time-frequency transform  
(ITU-T G.719)

Figure 5/JT-G719 は適応時間-周波数分解の全体処理を説明している。IsTransient フラグに基づいて、アルゴリズムは、窓掛け、時間域エリアシング、離散コサイン変換タイプ IV ( $DCT_{IV}$ ) の縦続接続による入力信号  $x(n)$  (ただし、 $n = 0, \dots, L-1$ ) の MLT か、4つのスペクトル  $y_0(k), y_1(k), y_2(k), y_3(k)$  (ただし、 $k = 0, \dots, L/4-1$ ) を持つ高精度な時間解像度の変換のどちらかを出力する。以降の節でその処理について説明する。

### 7. 2. 1 入力信号バッファリング、および窓掛け

過渡性の判定に使われるのと同じの入力フレーム信号  $x(n)$  は、そのフレームの周波数表現を計算するために使われる。この処理のタイムチャートを Figure 6/JT-G719 に示す。

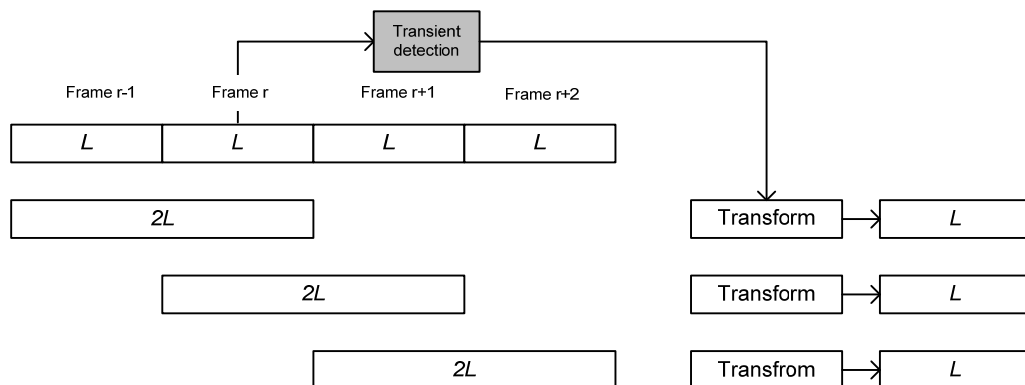


Figure 6/JT-G719 – Time chart of signal buffering and transform  
(ITU-T G.719)

20ms の入力信号  $x(n), n = 0, \dots, L-1$  は、前の 20ms フレームの信号  $x_{OLD}(n)$  とともに 40ms フレームを形成しながらバッファリングされる。このフレームは、Figure 6/JT-G719 に示されるように順に窓掛けされる。

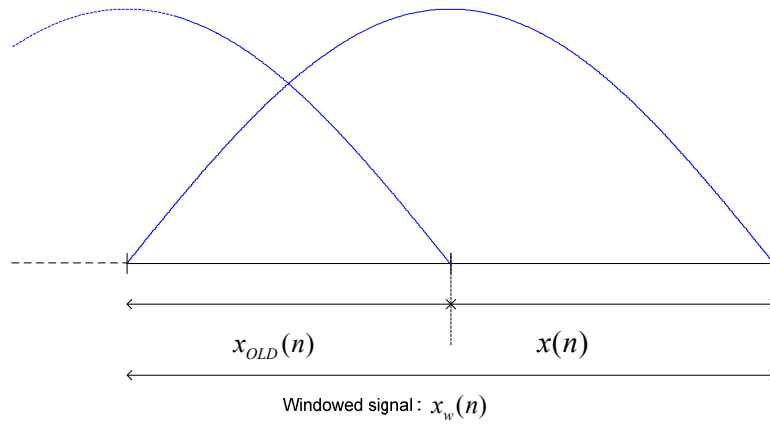


Figure 7/JT-G719 – Frame buffering and windowing  
(ITU-T G.719)

窓関数はサイン窓であり、次のように定義される。

$$h(n) = \sin\left[\left(n + \frac{1}{2}\right) \frac{\pi}{2L}\right], \quad n = 0, \dots, 2L-1 \quad (6)$$

結果的には下記のような窓掛けされた信号となる。

$$x_w(n) = \begin{cases} h(n)x_{OLD}(n), & n = 0, \dots, L-1 \\ h(n)x(n-L), & n = L, \dots, 2L-1 \end{cases} \quad (7)$$

### 7. 2. 2 時間域エリアシング

バッファリングと窓掛けをされたフレームは 40ms の長さ、つまり  $2L = 1920$  を持つ。それは時間域エリアシング(TDA)処理によって 20ms のフレームに変換される。

TDA 処理は次のような行列処理によって明示的に与えられる。

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{J}_{L/2} & -\mathbf{I}_{L/2} \\ \mathbf{I}_{L/2} & -\mathbf{J}_{L/2} & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}_w \quad (8)$$

ここで、行列  $\mathbf{I}_{L/2}$  と  $\mathbf{J}_{L/2}$  は  $L/2$  次の単位行列とその時間反転行列である。

$$\mathbf{I}_{L/2} = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}, \quad \mathbf{J}_{L/2} = \begin{bmatrix} 0 & & 1 \\ & \ddots & \\ 1 & & 0 \end{bmatrix} \quad (9)$$

ただし、(8)式の行列では行が列の半分であり、それゆえに出力信号  $\tilde{\mathbf{x}}(n)$  は入力信号  $x_w(n)$  の半分の長さとなる。この処理は Figure 8/JT-G719 に示される。



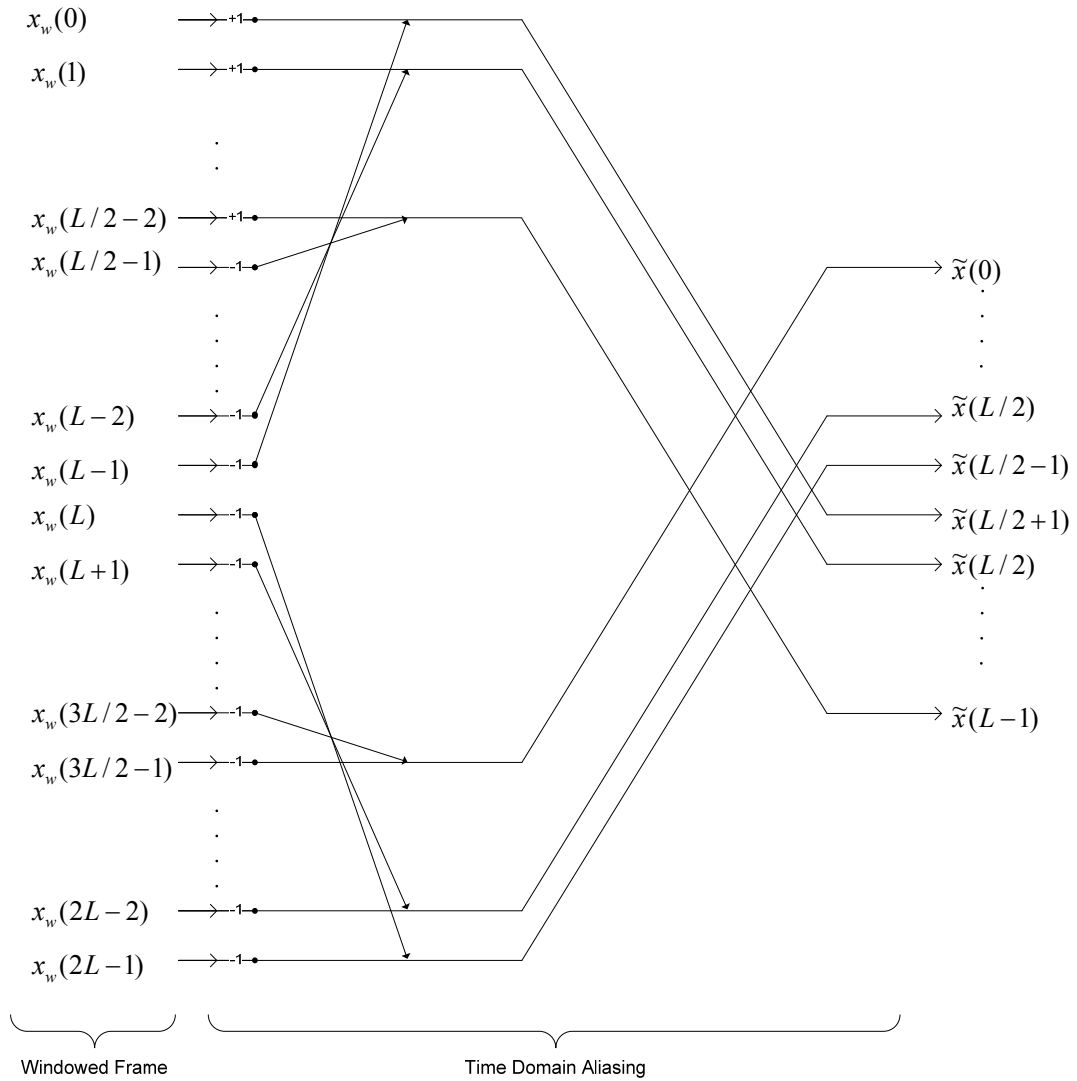


Figure 8/JT-G719 – Time domain aliasing operation  
(ITU-T G.719)

### 7. 2. 3 信号事前正規化

適応変換の固定小数点実装において小振幅信号に対して高い SNR を維持するために、各時間エリアシングフレーム  $\tilde{x}(n), n = 0, \dots, L-1$  は 32 ビット符号付整数のフルダイナミックレンジ、つまり  $[-2^{31}, \dots, 2^{31} - 1]$  に渡るように正規化される。

正規化は簡単な処理であり、時間域エリアシング信号のフレーム内の最大振幅サンプルを見つけて、次式からなる正規化インデックス  $Q_{\max}$  を計算する処理からなる。

$$Q_{\max} = 30 - \left\lceil \log_2 \left( \max_{n=0, \dots, L-1} |\tilde{x}(n)| \right) \right\rceil \quad (10)$$

アルゴリズムによる後続の処理が行われる前に、計算されたインデックス  $Q_{\max}$  はフレーム内の全てのサンプルを正規化するために利用される。この処理は以下の通りである。

$$\bar{\tilde{x}}(n) = 2^{Q_{\max}} \tilde{x}(n), n = 0, \dots, L-1 \quad (11)$$

以降、 $\bar{\tilde{x}}(n)$  の表記は簡略化のために省略する。

#### 7. 2. 4 定常モード

*IsTransient* フラグが *FALSE* の場合、タイプ IV の離散コサイン変換(DCT<sub>IV</sub>)が時間域エリアシング処理の出力に対して適用される。

長さ  $L$  の時間エリアシング信号の DCT<sub>IV</sub>[1]は次のように定義される。

$$y(k) = \sum_{n=0}^{L-1} \tilde{x}(n) \cos \left[ \left( n + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{L} \right], \quad k = 0, \dots, L-1 \quad (12)$$

定常モードの場合には、窓掛け、TDA、及び DCT<sub>IV</sub> が、変調重複変換(MLT)の適用と等価である。結果の信号  $y(k)$  は入力フレームのスペクトル係数となる。

#### 7. 2. 5 過渡モード

*IsTransient* フラグが *TRUE* の場合、時間域エリアシング信号のさらなる並べ替えが行われる。入力信号の時間的な整合を保つため、時間域エイリアシング処理の出力は、後続の処理を行う前に並べ替えられる必要がある。並べ替えは必要であり、並べ替えをしないと、結果として、フィルタバンクの基本機能によって時間と周波数の応答に不整合が生じることになる。この並べ替え処理は Figure 9/JT-G719 に示されており、TDA 出力信号  $\tilde{x}(n)$  の上半分と下半分を入れ換える処理からなる。

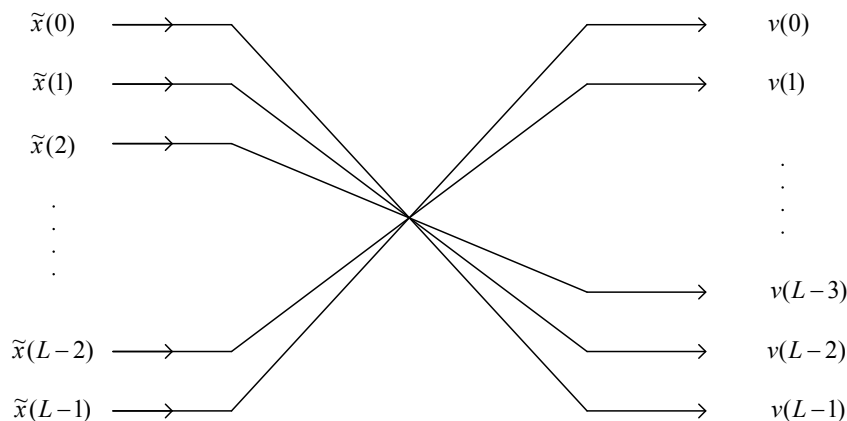


Figure 9/JT-G719 – Time domain aliasing re-ordering  
(ITU-T G.719)

$$v(n) = \tilde{x}(L-1-n), \quad n = 0, \dots, L-1 \quad (13)$$

この並べ替えは概念上だけのものであって、実際の計算は伴わない。信号  $v(n)$  に零詰めを行い、結果として生じる信号を4つの重複した同じ長さのサブフレームに分割することによって、より高い時間解像度が得られる。零詰めは信号の両端において  $L/8$  ずつ行われる。また、区間は50%ずつ重なり合い、各区間の長さは  $L/2$  である。2つの内側の区間は、長さ  $L/2$  のサイン窓を使って窓掛けされる。外側の区間は、サイン窓の半分を使って構築される。

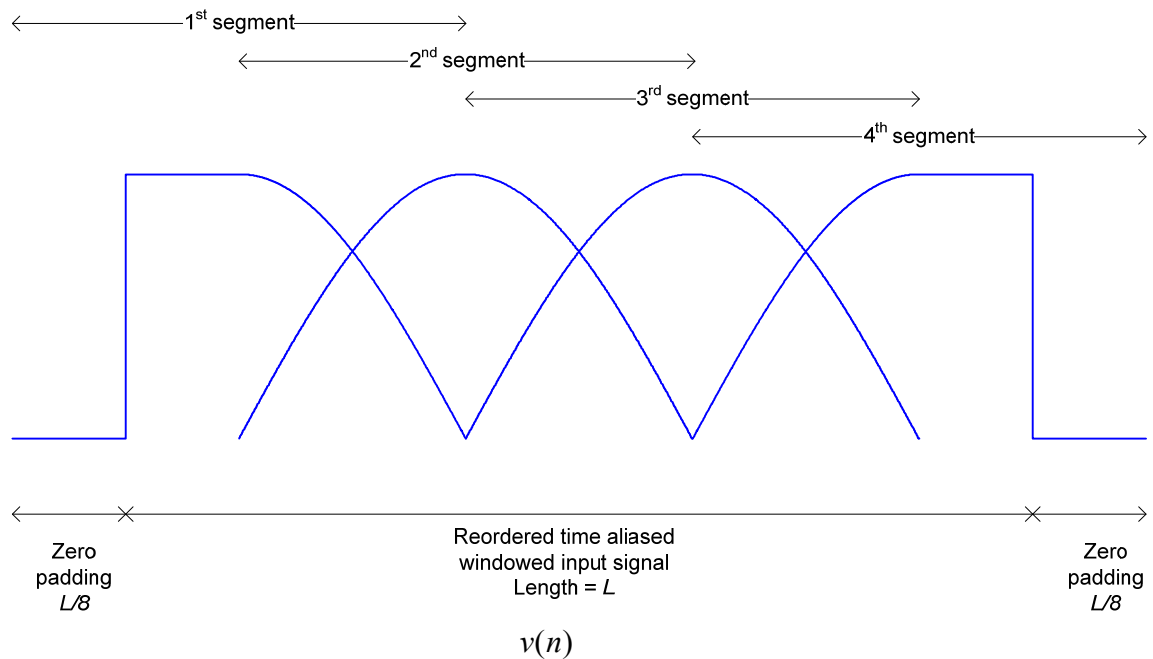


Figure 10/JT-G719 – Higher time resolution, division into four segments  
(ITU-T G.719)

この処理は Figure 10/JT-G719 に描かれる。それぞれの窓掛けされた区間は、さらに修正離散コサイン変換 (MDCT)(つまり、時間エリアシング+DCT<sub>IV</sub>)を適用することでさらなる処理が行われる。各区間の MDCT の出力は異なる時間の信号周波数を表現しており、過渡的な信号の場合には高い時間解像度が使われる。

4つの MDCT の各出力の長さは、入力区間の半分の長さ、すなわち  $L/4$  であり、それゆえ、この処理による追加の冗長は発生しない。

### 7. 2. 6 高速離散コサイン変換タイプIVアルゴリズム

DCT<sub>IV</sub> の計算は簡単かつ非常に効率的なアルゴリズムを使っている。アルゴリズムは、入力信号の半分の長さの FFT アルゴリズムの利用に基づいている。これは、定常モード、つまり入力信号の長さが  $L$  の場合には Figure 11/JT-G719 に示す簡略化した図によって説明される。過渡的な場合には、 $L$  を  $L/4$  に、 $\tilde{x}(n)$  と  $y(k)$  を  $\tilde{x}_m(n)$  と  $y_m(k)$  (ただし、 $m = 0,1,2,3$ ) に置き換えることで同じ図が適用できる。

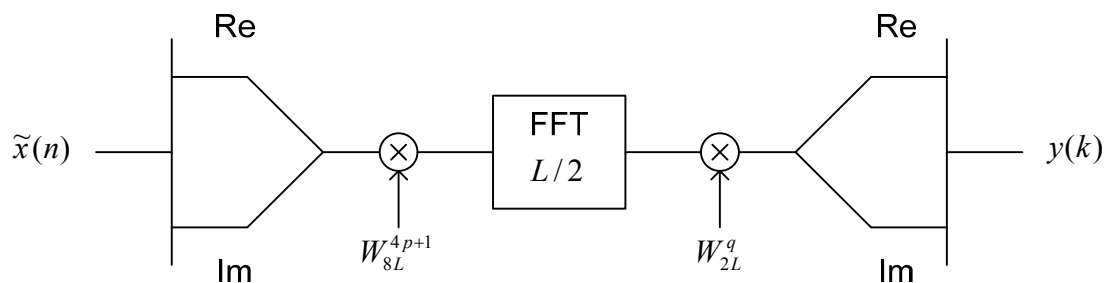


Figure 11/JT-G719 – Simplified diagram of the fast DCT<sub>IV</sub> algorithm  
(ITU-T G.719)

以降では、DCT<sub>IV</sub> アルゴリズムは定常モードとして説明されるが、アルゴリズム自体は過渡モードでも同一である。

アルゴリズム演算手順は以下の通りである。

a) 複素系列の形成

$$z(p) = \tilde{x}(2p) + j\tilde{x}(L-1-2p), \quad p = 0, \dots, L/2-1 \quad (14)$$

b) 複素事前乗算

形成された複素系列は次のように事前変調される。

$$\tilde{z}(p) = W_{8L}^{4p+1} z(p), \quad p = 0, \dots, L/2-1 \quad (15)$$

ここで、 $W_N = e^{j\frac{2\pi}{N}}$  は単位元の  $N$  乗根であり、それゆえ  $W_{8L}^{4p+1} = \cos\left(\frac{(4p+1)\pi}{4L}\right) + j\sin\left(\frac{(4p+1)\pi}{4L}\right)$  となる。

c) 複素逆 FFT 演算

次式により、係数  $\tilde{Z}(q)$  を得るために系列  $\tilde{z}(p)$ ,  $p = 0, \dots, L/2-1$  の正規化された複素逆フーリエ変換が計算される。

$$\tilde{Z}(q) = \sum_{p=0}^{L/2-1} \tilde{z}(p) W_{L/2}^{pq}, \quad q = 0, \dots, L/2-1 \quad (16)$$

定常モードの場合も過渡モードの場合も、Winograd 高速フーリエ変換アルゴリズムが使われる[2]。

d) 複素事後乗算

結果として得られる複素配列  $\tilde{Z}(q)$  (ただし、 $q = 0, \dots, L/2-1$ ) は次式に従って事後乗算される。

$$\bar{Z}(q) = W_{2L}^q \tilde{Z}(q), \quad q = 0, \dots, L/2-1 \quad (17)$$

e) 出力の形成

DCT<sub>IV</sub> ベクトルの最終的な出力は、次式によって、結果的に得られる配列  $\tilde{Z}(q)$  の実数部と虚数部を取ることによって構築される。

$$\begin{cases} y(2q) = \text{Re}\{\tilde{Z}(q)\} \\ y(L-1-2q) = -\text{Im}\{\tilde{Z}(q)\} \end{cases} \quad q = 0, \dots, L/2-1 \quad (18)$$

逆 DCT<sub>IV</sub> は DCT<sub>IV</sub> そのものなので、復号器での逆変換の演算でも同じアルゴリズムが使われる。

### 7. 2. 7 信号の事後正規化

得られたスペクトルは、実際は事前正規化された信号  $\tilde{x}(n)$  に対応するスペクトルであり、7.2.3 節の逆処理を適用することによって事後正規化される必要がある。これは、以下のように実施される。

$$y(k) = 2^{-Q_{\max}} \bar{y}(k), \quad k = 0, \dots, L-1 \quad (19)$$

ここで、 $\bar{y}(k)$  は、信号  $\tilde{x}(n)$  に対する変換を適用して得られるスペクトルに対応する。

### 7. 3 スペクトル係数のグルーピング

定常モードの処理では、スペクトル係数は4つのスペクトルグループに分割される。Table 5/JT-G719 に示す

ように、各スペクトルグループはそれぞれ長さの等しいサブベクトルに分割される。

Table 5/JT-G719 – Grouping of Spectral Coefficients in Stationary Mode  
(ITU-T G.719)

Group	Sub-vector length	Number of Sub-vectors	Number of Coefficients	Bandwidth (Hz)	Start (Hz)	End (Hz)
I	8	16	128	3 200	0	3 200
II	16	8	128	3 200	3 200	6 400
III	24	12	288	7 200	6 400	13 600
IV	32	8	256	6 400	13 600	20 000
<b>Total</b>		44	800	20 000		

サブベクトルの総数は  $P = 44$  である。サブベクトルの長さは各グループで異なり、周波数が高くなるほどサブベクトルの長さは長くなる。

このサブ分割は人間の耳をよく近似したスペクトル表現である。周波数が低いほど高い周波数解像度となり、周波数が高いほど低い周波数解像度となる。このグルーピングにより、量子化と符号化において利用できるビット割当てを効率的にできる。

グループ I は 16 個のサブベクトルから成り、各サブベクトルは 8 個の係数を持ち、3.2kHz 以下の周波数を表す。グループ II は 3.2-6.4kHz の範囲の周波数を表現し、16 個の係数のサブベクトルを 8 個持つ。グループ III では、24 個の係数を持つサブベクトルが 12 個あり、6.4-13.6kHz の周波数に相当する。グループ IV は、32 個の係数のサブベクトル 8 個で構成され、20kHz までの残りの周波数を表現する。20kHz 以上の周波数を表現するスペクトル係数は考慮されない。

Table 6, 7, 8, 9/JT-G719 は、定常モード時の各グループにおけるサブベクトルへのスペクトル係数の詳細な配置を示している。各テーブルで、 $p$  はサブベクトルのインデックスを表し、 $L_p$  はサブベクトルの長さを表し、 $s_p$  と  $e_p$  はサブベクトルを形成するスペクトル係数の開始インデックスと終了インデックスを表す。

Table 6/JT-G719 – Group I in Stationary Mode  
(ITU-T G.719)

Group	I																
	$p$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$L_p$		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
$s_p$		0	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120
$e_p$		7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127

Table 7/JT-G719 – Group II in Stationary Mode  
(ITU-T G.719)

Group	II							
$p$	16	17	18	19	20	21	22	23
$L_p$	16	16	16	16	16	16	16	16
$s_p$	128	144	160	176	192	208	224	240
$e_p$	143	159	175	191	207	223	239	255

Table 8/JT-G719 – Group III in Stationary Mode  
(ITU-T G.719)

Group	III											
$p$	24	25	26	27	28	29	30	31	32	33	34	35
$L_p$	24	24	24	24	24	24	24	24	24	24	24	24
$s_p$	256	280	304	328	352	376	400	424	448	472	496	520
$e_p$	279	303	327	351	375	399	423	447	471	495	519	543

Table 9/JT-G719 – Group IV in Stationary mode  
(ITU-T G.719)

Group	IV							
$p$	36	37	38	39	40	41	42	43
$L_p$	32	32	32	32	32	32	32	32
$s_p$	544	576	608	640	672	704	736	768
$e_p$	575	607	639	671	703	735	767	799

過渡モードの場合は、等価な4つの5ms変換の係数は、はじめインターリーブされ、その後上述のように4グループに分けられる。Table 10/JT-G719は、各変換に対応する係数がどのようにインデックスをつけられるかを示している。Table 11-14/JT-G719は、どのようにグループ I-IV の構造でインターリーブが行われるかを示す。

Table 10/JT-G719 – Numbering of coefficients of four 5-ms transforms  
(ITU-T G.719)

Coefficients of 1st 5-ms Transform			Coefficients of 2nd 5-ms Transform			Coefficients of 3rd 5-ms Transform			Coefficients of 4th 5-ms Transform		
#0	...	#239	#240	...	#479	#480	...	#719	#720	...	#959
$y_0(k)$			$y_1(k)$			$y_2(k)$			$y_3(k)$		

Table 11/JT-G719 – Group I in Transient Mode  
(ITU-T G.719)

Group	I															
$p$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$L_p$	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
$s_p$	0	8	240	248	480	488	720	728	16	24	256	264	496	504	736	744
$e_p$	7	15	247	255	487	495	727	735	23	31	263	271	503	511	743	751
$m$	0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3

Table 12/JT-G719 – Group II in Transient Mode  
(ITU-T G.719)

Group	II							
$p$	16	17	18	19	20	21	22	23
$L_p$	16	16	16	16	16	16	16	16
$s_p$	32	272	512	752	48	288	528	768
$e_p$	47	287	527	767	63	303	543	783
$m$	0	1	2	3	0	1	2	3

Table 13/JT-G719 – Group III in Transient Mode  
(ITU-T G.719)

Group	III												
$p$	24	25	26	27	28	29	30	31	32	33	34	35	
$L_p$	24	24	24	24	24	24	24	24	24	24	24	24	
$s_p$	64	304	544	784	88	328	568	808	112	352	592	832	
$e_p$	87	327	567	807	111	351	591	831	135	375	615	855	
$m$	0	1	2	3	0	1	2	3	0	1	2	3	

Table 14/JT-G719 – Group IV in Transient Mode  
(ITU-T G.719)

Group	IV							
$p$	36	37	38	39	40	41	42	43
$L_p$	32	32	32	32	32	32	32	32
$s_p$	136	376	616	856	168	408	648	888
$e_p$	167	407	647	887	199	439	679	919
$m$	0	1	2	3	0	1	2	3

#### 7.4 ノルム推定と量子化

過渡モードの場合、量子化ノルムは初めに順序付けされる。そのやり方は、効率的な差分ノルム符号化を考慮して、偶数サブフレームインデックス  $m=0,2$  に対応するノルムは、周波数域で昇順に並べ替えられ、一方、奇数サブフレームインデックス  $m=1,3$  のノルムは、周波数域で降順に並び替えられる。

サブベクトル  $N(p)$  のノルム、すなわちスペクトルエネルギーは、サブベクトルの二乗平均平方根(rms)の値として定義されて、以下の式に従って計算される。

$$N(p) = \sqrt{\frac{1}{L_p} \sum_{k=s_p}^{e_p} y(k)^2}, \quad p=0, \dots, P-1 \quad (20)$$

Table 15/JT-G719 – Codebook for norm quantization  
(ITU-T G.719)

index	Code	index	Code	index	Code	Index	Code
0	$2^{17.0}$	10	$2^{12.0}$	20	$2^{7.0}$	30	$2^{2.0}$
1	$2^{16.5}$	11	$2^{11.5}$	21	$2^{6.5}$	31	$2^{1.5}$
2	$2^{16.0}$	12	$2^{11.0}$	22	$2^{6.0}$	32	$2^{1.0}$
3	$2^{15.5}$	13	$2^{10.5}$	23	$2^{5.5}$	33	$2^{0.5}$
4	$2^{15.0}$	14	$2^{10.0}$	24	$2^{5.0}$	34	$2^{0.0}$
5	$2^{14.5}$	15	$2^{9.5}$	25	$2^{4.5}$	35	$2^{-0.5}$
6	$2^{14.0}$	16	$2^{9.0}$	26	$2^{4.0}$	36	$2^{-1.0}$
7	$2^{13.5}$	17	$2^{8.5}$	27	$2^{3.5}$	37	$2^{-1.5}$
8	$2^{13.0}$	18	$2^{8.0}$	28	$2^{3.0}$	38	$2^{-2.0}$
9	$2^{12.5}$	19	$2^{7.5}$	29	$2^{2.5}$	39	$2^{-2.5}$

各フレームにおいて、ノルムは、均一対数スカラ量子化器によって 3dB の 40 ステップで量子化されたスカラ値である。Table 15/JT-G719 は、対数量子化器のコードブックエントリを示している。

量子化ノルムのインデックス  $I_N(p)$  は、以下の式から容易に得ることができる。

$$I_N(p) = 34 - 2 \log_2 [N(p)] \quad \text{with } p = 0, \dots, P-1 \quad (21)$$

なお、上式は[0, 39]の範囲に制限されるように飽和させる。

最大量子化ノルムは、フレームに渡って探索される。32 kbit/s のレートでは、最大量子化ノルムがグループ



IVにあるか、もしくはグループ III にあり、かつグループ IV とグループ III との境界に近い位置にある場合に限り、グループ IV のスペクトル係数が符号化され、サブベクトルに対する閾値は  $p \geq 34$  と設定される。

フラグビット  $FlagL$  はグループ IV が符号化されているかどうかを示すために使われる。このフラグは復号器にサイド情報として送信される。

最低周波数サブベクトルの量子化インデックスすなわち  $I_N(0)$  は、直接復号器に送信される。残っている  $P-1=43$  サブベクトルにあるノルムの量子化インデックスは次式のように計算され、差分の符号化がなされる。

$$\Delta I_N(p) = I_N(p+1) - I_N(p), \quad p = 0, \dots, P-2 \quad (22)$$

差分インデックス  $\Delta I_N(p)$  は[-15, 16]の範囲へ拘束される。これは次に示すように、初めに負の差分インデックスを調整し、その後で正の差分インデックスを調整することで行われる。

1. 最高周波数サブベクトルから最低周波数サブベクトルまで順番に(22)式で定義された差分インデックスを計算する
2. If  $\Delta I_N(p) < -15$ ,  $I_N(p) = I_N(p+1) + 15$ ,  $p = 42, \dots, 0$
3. 最低周波数サブベクトルから最高周波数サブベクトルまで順番に差分インデックスを再計算する
4. If  $\Delta I_N(p) > 16$ ,  $\Delta I_N(p) = 16$  and  $I_N(p+1) = I_N(p) + 16$ ,  $p = 0, \dots, 42$
5. [0, 31]の範囲で調整された差分インデックスは  $\Delta I_N(p)$  に15の補正値を加えることによって得られる。

ハフマン符号化[5]が、調整された差分インデックスに対して適用される。この差分インデックスに対するハフマン符号は Table 16/JT-G719 で与えられる。

Table 16/JT-G719 – Huffman code for differential indices  
(ITU-T G.719)

Index	Code	Index	Code	Index	Code	Index	Code
0	0011010	8	001100	16	000	24	0011110
1	0111010	9	011100	17	010	25	0111110
2	1011010	10	101100	18	1010	26	1011110
3	1111010	11	111100	19	1110	27	1111110
4	0011011	12	0010	20	001110	28	0011111
5	0111011	13	0110	21	011110	29	0111111
6	1011011	14	100	22	101110	30	1011111
7	1111011	15	110	23	111110	31	1111111

フラグビット  $FlagN$  は符号化の種類を示すために用いられ、サイド情報として復号器に送信される。インデックスは、ノルムインデックス当たり 5bit でそのまま送信されるか、 $FlagN$  が設定されていればハフマン符号化されて送信される。このフラグ  $FlagN$  は、ハフマン符号化の結果から生じたビット数がインデックス数未満の場合に設定される。

最終結果として、再構築される量子化ノルムは  $I_N^q(p)$ ,  $p = 0, \dots, P-1$  として示される。

## 7. 5 ビット割り当て

### 7. 5. 1 ビット割り当てのためのノルム事後調整

心理音響的な重み付けやマスキング効果を考慮して、量子化されたノルムはビット割り当ての前に調整される。アルゴリズムは、スペクトル重み付け関数を用いた量子化ノルムのマッピングからなる。

最初に、量子化ノルムはスペクトル領域にマッピングされる。これは、定常信号の場合には量子化ノルムを

コピーし、遷移状態の場合には時間依存の4つのスペクトルの量子化ノルムを平均化することに相当する。これは、以下のように実施される。

$$Spe(p) = \begin{cases} \frac{1}{4} \sum_{k=0}^3 I_N(4 \lfloor p/4 \rfloor + k), & \text{if } IsTransient = TRUE \\ I_N(p), & \text{if } IsTransient = FALSE \end{cases} \quad (23)$$

得られたスペクトル群は、聴覚フィルタを通した耳の出力に対応した関数によってさらにマッピングされる。これは入力信号の心理音響的な重要度を表現している。この操作は以下の線形変換によって実行される。

$$BSpe(b) = \frac{1}{H_b} \sum_{p \in J_b} Spe(p) + T_b, \quad b = 0, \dots, B_{MAX} - 1 \quad (24)$$

ここで  $B_{MAX} = 20$  であり、定数  $H_b$  と  $T_b$  および加重インターバル  $J_b$  は Table17/JT-G719 によって与えられる。

Table 17/JT-G719 – Spectrum mapping function variables  
(ITU-T G.719)

$b$	$J_b$	$H_b$	$T_b$	$A(b)$
0	0	1	3	8
1	1	1	3	6
2	2	1	3	3
3	3	1	3	3
4	4	1	3	3
5	5	1	3	3
6	6	1	3	3
7	7	1	3	3
8	8	1	3	3
9	9	1	3	3
10	10, 11	2	4	3
11	12, 13	2	4	3
12	14, 15	2	4	3
13	16, 17	2	5	3
14	18, 19	2	5	3
15	20, 21, 22, 23	4	6	3
16	24, 25, 26	3	6	4
17	27, 28, 29	3	6	5
18	30, 31, 32, 33, 34	5	7	7
19	35, 36, 37, 38, 39, 40, 41, 42, 43	9	8	11

マッピングされたスペクトルは、次式に従って前方へ平滑化される。

$$BSpe(b) = \max(BSpe(b), BSpe(b-1) - 4), \quad b = 1, \dots, B_{MAX} \quad (25)$$

そして、結果である関数は次式に従って直接後方へ平滑化される。

$$BSpe(b) = \max(BSpe(b), BSpe(b+1) - 8), \quad b = B_{MAX} - 2, \dots, 0 \quad (26)$$

平滑化の後、聴覚における絶対閾値の平均レベルを考慮するために、結果の関数に閾値が設けられる。閾値の設定および再正規化は以下に従って実行される。

$$BSpe(b) = T_b - \max(BSpe(b), A(b)), \quad b = 0, \dots, B_{MAX} - 1 \quad (27)$$

ここで  $A(b)$  は Table17/JT-G719 で与えられ、聴覚における閾値を擬似的に表している。得られた関数は、スペクトルのダイナミックレンジに応じて圧縮、伸張され、以下の線形マッピングに応じて  $[-0, \dots, 3]$  の範囲にさらに適応的にマッピングされる。

$$BSpe(b) = \frac{4}{\max\{BSpe(b)\} - \min\{BSpe(b)\}} [BSpe(b) - \min\{BSpe(b)\}], \quad b = 0, \dots, B_{MAX} - 1 \quad (28)$$

得られたスペクトル群は、次式によってサブベクトル領域に逆マッピングされる。

$$WSpe(p) = BSpe(b), \text{ such as } p \in J_b, \quad p = 0, \dots, P-1 \quad (29)$$

変数  $IsTransient$  が  $TRUE$  にセットされる場合、すなわち遷移モードの場合は、得られたスペクトル群は以下に従ってさらに平滑化される。

$$WSpe(p) = \begin{cases} \frac{1}{4} \sum_{k=0}^3 WSpe(4 \lfloor p/4 \rfloor + k), & \text{if } IsTransient = TRUE \\ WSpe(p), & \text{if } IsTransient = FALSE \end{cases} \quad (30)$$

最終的に、ビット割り当てに用いられるノルムが以下のように計算される。

$$\tilde{I}_N^q(p) = I_N^q(p) + WSpe(p) \quad (31)$$

### 7. 5. 2 ノルムに基づくビット割り当て

適応ビット割り当ての手法は、サブベクトル間のフレームの利用可能なビットに割り当てるために調整された量子化ノルム  $\tilde{I}_N^q(p), p = 0, \dots, P-1$  を用いる。

各正規化変換係数に割り当てられる最大ビット数は、初期値  $R_{\max} = 9$  に設定される。

最初に、ビット割り当てベクトルの各要素、つまりサブベクトルに対する各サンプルあたりのビット割り当ては0ビットとする。これは以下のように実行される。

$$R(p) = 0 \quad \text{for } p = 0, \dots, P-1 \quad (32)$$

$\Omega$  で表記される残余ビット数は、総割り当て可能ビット数  $\Psi$  にセットされる。後者 ( $\Psi$ ) は、20ms フレームにおける総割り当て可能ビット数から、フラグビット数、ノルム符号ビット数、存在するならば雑音レベルビット数 (7.7 および 7.8 節参照) を減じたものである。

ビット割り当てベクトル  $R(p), p = 0, \dots, P-1$ 、残余ビット数  $\Omega$  および  $\Psi$  は下式が成り立つ。

$$\sum_{p=0}^{P-1} R(p)L_p + \Omega = \Psi \quad (33)$$

各反復において、44 のサブベクトルの中の最大ノルムを持つサブベクトルのインデックス  $p_{\max}$  は、以下のよう  
に得られる。

$$p_{\max} = \arg \max_{p=0, \dots, P-1} [\tilde{I}_N^q(p)] \quad (34)$$

このサブベクトルに対して、アルゴリズムは各スペクトル係数に1ビットずつ割り当てていく。すなわち、 $R(p_{\max})$  は1  
ずつインクリメントされる。一方で、ノルムは6 dB ずつデクリメントされる。すなわち、ノルム  
インデックスは2 ずつデクリメントされる。これら二つの操作は以下のように実行される。

$$\begin{aligned} R(p_{\max}) &\leftarrow R(p_{\max}) + 1 \\ \tilde{I}_N^q(p_{\max}) &\leftarrow \tilde{I}_N^q(p_{\max}) - 2 \end{aligned} \quad (35)$$

ビット割り当てベクトルの更新にしたがって、残余ビット数  $\Omega$  も合わせて更新される。

$$\Omega \leftarrow \Omega - L_{p_{\max}} \quad (36)$$

ビット割り当て  $R(p_{\max})$  が最大許容ビットレート  $R_{\max}$  に到達したら、次の反復においてそのベクトルが対象  
とならないようノルムを負の無限大（実際には  $-2^{15}$ ）とする。これらの操作は、 $\Omega$  が  $L_{p_{\max}}$  未満となるまで繰  
り返される。

反復手順が終了したとき、残余ビットの数に依存して、ビット割り当てループを終了させた  $L_{p_{\max}}$  よりも低次  
元のサブベクトルに対し残りのビットが割り当てられる。

レートが 32 kbit/s のとき、*FlagL* が設定されていなければ、利用可能ビットは始めから 36 個までのサブベク  
トルにしか割り当てられない。

## 7. 6 高速格子ベクトル量子化

各サブベクトル  $p$  において、変換係数は量子化ノルム  $I_N^q(p)$ 、 $p=0, \dots, P-1$  によって正規化される。これら  
正規化された係数は、一つあるいはいくつかの8次元係数のベクトルから成る。各サブベクトル  $p$  に対し、8  
次元ベクトルは  $\mathbf{y}_{p,l}$ 、 $l=1, \dots, L_p/8$  と表記される。

各  $\mathbf{y}_{p,l}$  は、その後高速格子ベクトル量子化 (FLVQ) 手法によって量子化される。この量子化器は二つの  
量子化器から成る。一つはLVQ1と呼ばれる、 $RE_8$ に基づく低レート格子ベクトル量子化器であり、もう一  
つはLVQ2と呼ばれる、 $D_8$ に基づく高レート格子ベクトル量子化器[3]である。ビット割り当てはサンプル毎  
に実行されるので、サブベクトル  $p$  に属する全ての8次元ベクトル  $\mathbf{y}_{p,l}$  は、同じビット割り当て  $R(p)$  が割り当  
てられる。

サブベクトルの各係数に対し1ビットが割り当てられているとき、低レート量子化器LVQ1 (7.6.1節参照)  
は、サブベクトルの正規化された係数を量子化するために用いられ、そうでない場合は、係数は高レート量子  
化器LVQ2 (7.6.2節参照) によって量子化される。量子化符号帳は有限格子領域から構成され、正規化され  
た係数の確率密度関数 (PDF) にマッチする。符号ベクトルは、簡単な代数的方法によって生成され番号付  
けされる。

### 7. 6. 1 $RE_8$ に基づく格子ベクトル量子化

格子  $RE_8$  は、原点を中心とした半径  $2\sqrt{2}r$ 、 $r=0,1,2, \dots$  の同心球上の点から成る。球上の点の集まりが球状符  
号を構成し、量子化符号帳として用いられる[4]。

LVQ1において、符号帳は  $r=1$  の球上のある全240点の  $RE_8$  と、格子  $RE_8$  に属さない追加の16点から成る。

追加の点は、ベクトル  $(-2, 0, 0, 0, 0, 0, 0, 0)$  と  $(2, 0, 0, 0, 0, 0, 0, 0)$  の順列から得られ、原点に近い入力ベクトルの量子化に用いられる。番号付けアルゴリズムを高速化するために、符号長の符号ベクトルが特定の順に用意され、これは Table18/JT-G719 に示される。

レート  $R(p)=1$  に対し、各 8 次元の係数ベクトル  $\mathbf{y}_{p,l} = (y_{p,l}^1, y_{p,l}^2, \dots, y_{p,l}^8)$  は以下のように量子化される。

- 1) オフセット  $a=2^{-6}$  をベクトル  $\mathbf{y}_{p,l}$  の各要素に適用する：

$$\bar{\mathbf{y}}_{p,l} = \mathbf{y}_{p,l} - \mathbf{a} \quad (37)$$

- 2) スケーリングファクタ  $\beta$  によってベクトル  $\bar{\mathbf{y}}_{p,l}$  をスケーリングする：

$$\hat{\mathbf{y}}_{p,l} = \beta \bar{\mathbf{y}}_{p,l} \quad (38)$$

最適なスケーリングファクタとして経験的に  $\beta=1.10$  が選ばれる。

- 3)  $\hat{\mathbf{y}}_{p,l}$  の各要素を降順に並べ替えて新たなベクトル  $\hat{\mathbf{y}}'_{p,l}$  を得る。
- 4) Table19/JT-G719 から、平均二乗誤差(MSE)において  $\hat{\mathbf{y}}'_{p,l}$  に最も近いベクトル  $\mathbf{v}$  を見つける。Table19/JT-G719 で与えられるベクトルは、符号ベクトルにおけるリーダと呼ばれ、符号帳の任意のベクトルはこれらの代表の順列により生成される。
- 5)  $\mathbf{v}$  の要素を元の順に並べ替えて最適符号ベクトル  $\mathbf{c}_{p,l}$  を得る。
- 6)  $\mathbf{c}_{p,l}$  のインデックス  $k$  を算出して終了する。

Table 18/JT-G719 – Codebook of the lower-rate lattice vector quantizer (LVQ1)  
(ITU-T G.719)

Index	Codeword	Index	Codeword	Index	Codeword	Index	Codeword
0	-2 0 0 0 0 0 0 0	64	0 0 0 -2 0 0 2 0	128	-1 -1 1 1 1 1 1 1	192	1 1 1 -1 1 -1 -1 -1
1	0 -2 0 0 0 0 0 0	65	0 0 0 -2 0 0 0 2	129	-1 1 -1 1 1 1 1 1	193	1 1 1 -1 -1 1 -1 -1
2	0 0 -2 0 0 0 0 0	66	0 0 0 0 -2 2 0 0	130	-1 1 1 -1 1 1 1 1	194	1 1 1 -1 -1 -1 1 -1
3	0 0 0 -2 0 0 0 0	67	0 0 0 0 -2 0 2 0	131	-1 1 1 1 -1 1 1 1	195	1 1 1 -1 -1 -1 -1 1
4	0 0 0 0 -2 0 0 0	68	0 0 0 0 -2 0 0 2	132	-1 1 1 1 1 -1 1 1	196	1 1 -1 1 1 -1 -1 -1
5	0 0 0 0 0 -2 0 0	69	0 0 0 0 0 -2 2 0	133	-1 1 1 1 1 1 -1 1	197	1 1 -1 1 -1 1 -1 -1
6	0 0 0 0 0 0 -2 0	70	0 0 0 0 0 -2 0 2	134	-1 1 1 1 1 1 1 -1	198	1 1 -1 1 -1 -1 1 -1
7	0 0 0 0 0 0 0 -2	71	0 0 0 0 0 0 -2 2	135	1 -1 -1 1 1 1 1 1	199	1 1 -1 1 -1 -1 -1 1
8	2 0 0 0 0 0 0 0	72	2 -2 0 0 0 0 0 0	136	1 -1 1 -1 1 1 1 1	200	1 1 -1 -1 1 1 -1 -1
9	0 2 0 0 0 0 0 0	73	2 0 -2 0 0 0 0 0	137	1 -1 1 1 -1 1 1 1	201	1 1 -1 -1 1 -1 1 -1
10	0 0 2 0 0 0 0 0	74	2 0 0 -2 0 0 0 0	138	1 -1 1 1 1 -1 1 1	202	1 1 -1 -1 1 -1 -1 1
11	0 0 0 2 0 0 0 0	75	2 0 0 0 -2 0 0 0	139	1 -1 1 1 1 1 -1 1	203	1 1 -1 -1 -1 -1 1 1
12	0 0 0 0 2 0 0 0	76	2 0 0 0 0 -2 0 0	140	1 -1 1 1 1 1 1 -1	204	1 1 -1 -1 -1 1 -1 1
13	0 0 0 0 0 2 0 0	77	2 0 0 0 0 0 -2 0	141	1 1 -1 -1 1 1 1 1	205	1 1 -1 -1 -1 1 1 -1
14	0 0 0 0 0 0 2 0	78	2 0 0 0 0 0 0 -2	142	1 1 -1 1 -1 1 1 1	206	1 -1 1 1 1 -1 -1 -1
15	0 0 0 0 0 0 0 2	79	0 2 -2 0 0 0 0 0	143	1 1 -1 1 1 -1 1 1	207	1 -1 1 1 -1 1 -1 -1
16	-2 -2 0 0 0 0 0 0	80	0 2 0 -2 0 0 0 0	144	1 1 -1 1 1 1 -1 1	208	1 -1 1 1 -1 -1 1 -1
17	-2 0 -2 0 0 0 0 0	81	0 2 0 0 -2 0 0 0	145	1 1 -1 1 1 1 1 -1	209	1 -1 1 1 -1 -1 -1 1
18	-2 0 0 -2 0 0 0 0	82	0 2 0 0 0 -2 0 0	146	1 1 1 -1 -1 1 1 1	210	1 -1 1 -1 1 1 -1 -1
19	-2 0 0 0 -2 0 0 0	83	0 2 0 0 0 0 -2 0	147	1 1 1 -1 1 -1 1 1	211	1 -1 1 -1 1 -1 1 -1
20	-2 0 0 0 0 -2 0 0	84	0 2 0 0 0 0 0 -2	148	1 1 1 -1 1 1 -1 1	212	1 -1 1 -1 1 -1 -1 1
21	-2 0 0 0 0 0 -2 0	85	0 0 2 -2 0 0 0 0	149	1 1 1 -1 1 1 1 -1	213	1 -1 1 -1 -1 -1 1 1
22	-2 0 0 0 0 0 0 -2	86	0 0 2 0 -2 0 0 0	150	1 1 1 1 -1 -1 1 1	214	1 -1 1 -1 -1 1 -1 1
23	0 -2 -2 0 0 0 0 0	87	0 0 2 0 0 -2 0 0	151	1 1 1 1 -1 1 -1 1	215	1 -1 1 -1 -1 1 1 -1
24	0 -2 0 -2 0 0 0 0	88	0 0 2 0 0 0 -2 0	152	1 1 1 1 -1 1 1 -1	216	1 -1 -1 -1 -1 1 1 1
25	0 -2 0 0 -2 0 0 0	89	0 0 2 0 0 0 0 -2	153	1 1 1 1 1 -1 -1 1	217	1 -1 -1 -1 1 -1 1 1
26	0 -2 0 0 0 -2 0 0	90	0 0 0 2 -2 0 0 0	154	1 1 1 1 1 -1 1 -1	218	1 -1 -1 -1 1 1 -1 1
27	0 -2 0 0 0 0 -2 0	91	0 0 0 2 0 -2 0 0	155	1 1 1 1 1 1 -1 -1	219	1 -1 -1 -1 1 1 1 -1
28	0 -2 0 0 0 0 0 -2	92	0 0 0 2 0 0 -2 0	156	-1 -1 -1 -1 1 1 1 1	220	1 -1 -1 1 1 1 -1 -1

Index	Codeword	Index	Codeword	Index	Codeword	Index	Codeword
29	0 0 -2 -2 0 0 0 0	93	0 0 0 2 0 0 0 -2	157	-1 -1 -1 1 -1 1 1 1	221	1 -1 -1 1 1 -1 1 -1
30	0 0 -2 0 -2 0 0 0	94	0 0 0 0 2 -2 0 0	158	-1 -1 -1 1 1 -1 1 1	222	1 -1 -1 1 1 -1 -1 1
31	0 0 -2 0 0 -2 0 0	95	0 0 0 0 2 0 -2 0	159	-1 -1 -1 1 1 1 -1 1	223	1 -1 -1 1 -1 -1 1 1
32	0 0 -2 0 0 0 -2 0	96	0 0 0 0 2 0 0 -2	160	-1 -1 -1 1 1 1 1 -1	224	1 -1 -1 1 -1 1 -1 1
33	0 0 -2 0 0 0 0 -2	97	0 0 0 0 0 2 -2 0	161	-1 -1 1 -1 -1 1 1 1	225	1 -1 -1 1 -1 1 1 -1
34	0 0 0 -2 -2 0 0 0	98	0 0 0 0 0 2 0 -2	162	-1 -1 1 -1 1 -1 1 1	226	1 1 -1 -1 -1 -1 -1 -1
35	0 0 0 -2 0 -2 0 0	99	0 0 0 0 0 0 2 -2	163	-1 -1 1 -1 1 1 -1 1	227	1 -1 1 -1 -1 -1 -1 -1
36	0 0 0 -2 0 0 -2 0	100	2 2 0 0 0 0 0 0	164	-1 -1 1 -1 1 1 1 -1	228	1 -1 -1 1 -1 -1 -1 -1
37	0 0 0 -2 0 0 0 -2	101	2 0 2 0 0 0 0 0	165	-1 -1 1 1 -1 -1 1 1	229	1 -1 -1 -1 1 -1 -1 -1
38	0 0 0 0 -2 -2 0 0	102	2 0 0 2 0 0 0 0	166	-1 -1 1 1 -1 -1 1 -1	230	1 -1 -1 -1 -1 1 -1 -1
39	0 0 0 0 -2 0 -2 0	103	2 0 0 0 2 0 0 0	167	-1 -1 1 1 -1 1 1 -1	231	1 -1 -1 -1 -1 -1 1 -1
40	0 0 0 0 -2 0 0 -2	104	2 0 0 0 0 2 0 0	168	-1 -1 1 1 1 1 -1 -1	232	1 -1 -1 -1 -1 -1 1 1
41	0 0 0 0 0 -2 -2 0	105	2 0 0 0 0 0 2 0	169	-1 -1 1 1 1 -1 1 -1	233	-1 1 1 -1 -1 -1 -1 -1
42	0 0 0 0 0 -2 0 -2	106	2 0 0 0 0 0 0 2	170	-1 -1 1 1 1 -1 -1 1	234	-1 1 1 -1 1 -1 -1 -1
43	0 0 0 0 0 0 -2 -2	107	0 2 2 0 0 0 0 0	171	-1 1 -1 -1 -1 1 1 1	235	-1 1 -1 -1 1 -1 -1 -1
44	-2 2 0 0 0 0 0 0	108	0 2 0 2 0 0 0 0	172	-1 1 -1 -1 1 -1 1 1	236	-1 1 -1 -1 -1 1 -1 -1
45	-2 0 2 0 0 0 0 0	109	0 2 0 0 2 0 0 0	173	-1 1 -1 -1 1 1 -1 1	237	-1 1 -1 -1 -1 -1 1 -1
46	-2 0 0 2 0 0 0 0	110	0 2 0 0 0 2 0 0	174	-1 1 -1 -1 1 1 1 -1	238	-1 1 -1 -1 -1 -1 1 1
47	-2 0 0 0 2 0 0 0	111	0 2 0 0 0 0 2 0	175	-1 1 -1 1 -1 -1 1 1	239	-1 -1 1 1 -1 -1 -1 -1
48	-2 0 0 0 0 2 0 0	112	0 2 0 0 0 0 0 2	176	-1 1 -1 1 -1 1 -1 1	240	-1 -1 1 -1 1 -1 -1 -1
49	-2 0 0 0 0 0 2 0	113	0 0 2 2 0 0 0 0	177	-1 1 -1 1 -1 1 1 -1	241	-1 -1 1 -1 -1 1 -1 -1
50	-2 0 0 0 0 0 0 2	114	0 0 2 0 2 0 0 0	178	-1 1 -1 1 1 1 -1 -1	242	-1 -1 1 -1 -1 -1 1 -1
51	0 -2 2 0 0 0 0 0	115	0 0 2 0 0 2 0 0	179	-1 1 -1 1 1 -1 -1 -1	243	-1 -1 1 -1 -1 -1 1 1
52	0 -2 0 2 0 0 0 0	116	0 0 2 0 0 0 2 0	180	-1 1 -1 1 1 -1 -1 1	244	-1 -1 -1 1 1 -1 -1 -1
53	0 -2 0 0 2 0 0 0	117	0 0 2 0 0 0 0 2	181	-1 1 1 1 1 1 -1 -1	245	-1 -1 -1 1 1 -1 -1 -1
54	0 -2 0 0 0 2 0 0	118	0 0 0 2 2 0 0 0	182	-1 1 1 1 -1 -1 1 -1	246	-1 -1 -1 1 -1 -1 1 -1
55	0 -2 0 0 0 0 2 0	119	0 0 0 2 2 0 2 0	183	-1 1 1 1 -1 -1 1 -1	247	-1 -1 -1 1 -1 -1 1 1
56	0 -2 0 0 0 0 0 2	120	0 0 0 2 0 0 2 0	184	-1 1 1 1 1 -1 -1 1	248	-1 -1 -1 -1 1 1 -1 -1
57	0 0 -2 2 0 0 0 0	121	0 0 0 2 0 0 0 2	185	-1 1 1 -1 -1 -1 1 1	249	-1 -1 -1 -1 1 -1 1 -1
58	0 0 -2 0 2 0 0 0	122	0 0 0 0 2 2 0 0	186	-1 1 1 -1 -1 1 -1 1	250	-1 -1 -1 -1 1 -1 1 1
59	0 0 -2 0 0 2 0 0	123	0 0 0 0 2 0 2 0	187	-1 1 1 -1 -1 1 1 -1	251	-1 -1 -1 -1 -1 1 1 -1
60	0 0 -2 0 0 0 2 0	124	0 0 0 0 2 0 0 2	188	-1 1 1 -1 1 1 -1 -1	252	-1 -1 -1 -1 -1 -1 1 1
61	0 0 -2 0 0 0 0 2	125	0 0 0 0 0 2 2 0	189	-1 1 1 -1 1 -1 1 -1	253	-1 -1 -1 -1 -1 -1 1 1
62	0 0 0 -2 2 0 0 0	126	0 0 0 0 0 2 0 2	190	-1 1 1 -1 1 -1 -1 1	254	-1 -1 -1 -1 -1 -1 -1 -1
63	0 0 0 -2 0 2 0 0	127	0 0 0 0 0 0 2 2	191	1 1 1 1 -1 -1 -1 -1	255	1 1 1 1 1 1 1 1

Table 19/JT-G719 – Leaders of the code-vectors of LVQ1 (ITU-T G.719)

Index	Leader
0	0 0 0 0 0 0 0 -2
1	2 0 0 0 0 0 0 0
2	0 0 0 0 0 0 -2 -2
3	2 0 0 0 0 0 0 -2
4	2 2 0 0 0 0 0 0
5	1 1 1 1 1 1 -1 -1
6	1 1 1 1 -1 -1 -1 -1
7	1 1 -1 -1 -1 -1 -1 -1
8	-1 -1 -1 -1 -1 -1 -1 -1
9	1 1 1 1 1 1 1 1

7. 6. 2  $D_8$ に基づく格子ベクトル量子化

格子  $D_8$  はよく知られた格子であり以下で定義される[3].

$$D_8 = \left\{ (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7) \in Z_8 \mid \sum_{i=0}^7 y_i = \text{even} \right\} \quad (39)$$

ここで  $Z_8$  は全てが整数座標点から成る格子である。  $D_8$  は整数格子であり、成分の合計が偶数となる整数座標  $\mathbf{y} = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$  から成ることが分かる。

係数あたりに与えられたビット数、即ちレート  $R(p)$  ( $2 \leq R(p) \leq 9$ ) に対して、8次元係数ベクトル  $\mathbf{y}_{p,l} = (y_{p,l}^1, y_{p,l}^2, \dots, y_{p,l}^8)$  はLVQ2によって次のように量子化される。

- 1) オフセット  $a = 2^{-6}$  をベクトル  $\mathbf{y}_{p,l}$  の各要素に適用する：

$$\bar{\mathbf{y}}_{p,l} = \mathbf{y}_{p,l} - \mathbf{a} \quad (40)$$

打ち切りボロノイ領域の境界上に格子点があるのを回避するためである。

- 2) スケーリングファクタ  $\beta(p)$  によってベクトル  $\bar{\mathbf{y}}_{p,l}$  を次のようにスケーリングする

$$\hat{\mathbf{y}}_{p,l} = \beta(p) \bar{\mathbf{y}}_{p,l} \quad (41)$$

与えられた  $R(p)$  に対し、最適なスケーリングファクタは経験的に  $\beta(p) = 2^{R(p)} / 6$  が選択される。

- 3) スケーリングされたベクトル  $\hat{\mathbf{y}}_{p,l}$  に最も近い  $D_8$  の格子点  $\mathbf{v}$  を見つける。これは[3]で記述された探索アルゴリズムを用いて実行される。

- 4)  $\mathbf{v}$  を与えられたレート  $R(p)$  によって打ち切られたボロノイ領域中の符号ベクトルとみなし  $\mathbf{v}$  のインデックスベクトル  $\mathbf{k} = (k_1, k_2, \dots, k_8)$  を計算する。ここで  $0 \leq k_i \leq 2^{R(p)} - 1$ 、 $i = 1, \dots, 8$  である。インデックス  $\mathbf{k}$  は以下で与えられる。

$$\mathbf{k} = \mathbf{v} \mathbf{G}^{-1} \bmod r \quad \text{with } r = 2^{R(p)} \quad (42)$$

ここで  $\mathbf{G}$  は  $D_8$  に対する生成行列で、以下で定義される[3]：

$$\mathbf{G} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)$$

および、

$$\mathbf{G}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (44)$$

- 5) [3]に記述されるアルゴリズムでインデックスベクトル  $\mathbf{k}$  から符号ベクトル  $\mathbf{c}$  を計算し、得られた  $\mathbf{c}$  を  $\mathbf{v}$  と比較する。  $\mathbf{c}$  と  $\mathbf{v}$  が一致するならば、  $\mathbf{k}$  が  $\hat{\mathbf{y}}_{p,l}$  に対する最適符号ベクトルであり、処理を終了する。一致しなければ、入力ベクトル  $\hat{\mathbf{y}}_{p,l}$  は外れ値であり以下で量子化される。
- 6) ベクトル  $\hat{\mathbf{y}}_{p,l}$  を 2 で割って縮小する：  $\hat{\mathbf{y}}_{p,l} = \hat{\mathbf{y}}_{p,l} / 2$
- 7) 新しくスケールリングされた  $\hat{\mathbf{y}}_{p,l}$  に最も近い  $D_8$  の格子点  $\mathbf{u}$  を見つけ、  $\mathbf{u}$  のインデックスベクトル  $\mathbf{j}$  を算出する。
- 8) インデックスベクトル  $\mathbf{j}$  から符号ベクトル  $\mathbf{c}$  を見つけ、  $\mathbf{c}$  と  $\mathbf{u}$  を比較する。  $\mathbf{c}$  と  $\mathbf{u}$  が一致しなければ、ステップ 6) から 8) までは繰り返す。一致するとき、  $\mathbf{w} = \hat{\mathbf{y}}_{p,l} / 16$  を計算する。ここで変換係数は正規化されているため、打ち切り格子における外れ値に対して符号ベクトルを探す操作は多くの反復を必要としない。
- 9)  $\hat{\mathbf{y}}_{p,l} = \hat{\mathbf{y}}_{p,l} + \mathbf{w}$  を計算する。
- 10)  $\hat{\mathbf{y}}_{p,l}$  に最も近い  $D_8$  の格子点  $\mathbf{u}$  を見つけ、  $\mathbf{u}$  のインデックスベクトル  $\mathbf{j}$  を算出する。
- 11) インデックスベクトル  $\mathbf{j}$  から符号ベクトル  $\mathbf{c}$  を見つけ、  $\mathbf{c}$  と  $\mathbf{u}$  を比較する。  $\mathbf{c}$  と  $\mathbf{u}$  が厳密に等しければ、  $\mathbf{k} = \mathbf{j}$  としステップ 9) から 11) までは繰り返す。等しくなければ、  $\mathbf{k}$  を  $\hat{\mathbf{y}}_{p,l}$  に対する最適符号ベクトルのインデックスとし、処理を終了する。

各サブベクトルにおける高レート格子ベクトル量子化 (LVQ2) のインデックスに対して、オプションとしてハフマン符号が使われる。全てのグループにとって必要な総ビットが加えられる。ハフマン符号によるビット数が割り当てられたビット数より少ないとき、  $R(p) \leq 5$  での量子化インデックスに対しハフマン符号が適用され、ハフマン符号フラグ *FlagC* がセットされる。そして、節約されたビットは残余ビットに割り当てられる。ハフマン符号によるビット数が割り当てられた固定ビット数以上のとき、ハフマン符号は用いられず、ハフマン符号フラグ *FlagC* はクリアされる。どちらの場合においても、ハフマン符号フラグは補助情報として復号器に伝達される。量子化インデックスに対するハフマン符号は Table20/JT-G719 で与えられる。

Table 20/JT-G719 – Huffman code for LVQ2 quantization indices  
(ITU-T G.719)

LVQ2 Index	Huffman code			
	2-b LVQ2	3-b LVQ2	4-b LVQ2	5-b LVQ2
0	0	00	00	00
1	110	01	110	010
2	111	1001	0110	1000
3	10	10000	0111	10100
4		10001	10100	10101
5		1010	10101	110000
6		1011	10110	110001
7		11	101110	110010
8			101111	110011
9			10000	1110000
10			10001	1110001
11			10010	1110010
12			10011	1110011
13			0100	1110100
14			0101	1110101



LVQ2 Index	Huffman code			
	2-b LVQ2	3-b LVQ2	4-b LVQ2	5-b LVQ2
15			111	1110110
16				1110111
17				1111000
18				1111001
19				1111010
20				1111011
21				1111100
22				1111101
23				1111111
24				110100
25				110101
26				110110
27				110111
28				10110
29				10111
30				1001
31				011

最終的に、残余ビットは、適応ビット割り当てによるビットが割り当てられなかったサブベクトルにおける係数の量子化に用いられる。

### 7. 7 雑音レベル調整

ビット割り当てにおいて 1 ビットも与えられなかったサブベクトルに属するスペクトル係数は符号化されない。これは、全ての変換係数が復号器に伝送されるわけではないことを意味する。

*IsTransient=FALSE* のとき、つまり定常状態のフレームにおいてのみ、これら符号化されなかったスペクトル係数のレベルが符号化器において評価、量子化される。

量子化されていない信号レベルの評価は、正規化されたスペクトル領域において直接行われる。雑音レベルの評価に先立って、雑音充填領域と帯域拡張領域間の遷移周波数が評価される。遷移周波数は符号化器と復号器において同様に評価され、雑音充填の終わりと帯域拡張の始まりを示す。

遷移周波数は、最後に量子化された帯域を基準として評価される。一般的な方法は、P-1 から始めて 0 までの全帯域を調べることである。現在の帯域に量子化された係数がなければ、帯域拡張領域になるものとして印される。その帯域に量子化された係数が存在すれば、この帯域および後に続く帯域は雑音充填領域になる。Figure12 / JT-G719 にこの過程が図示されている。ここで、遷移周波数は、帯域拡張と雑音充填との間を分割する。

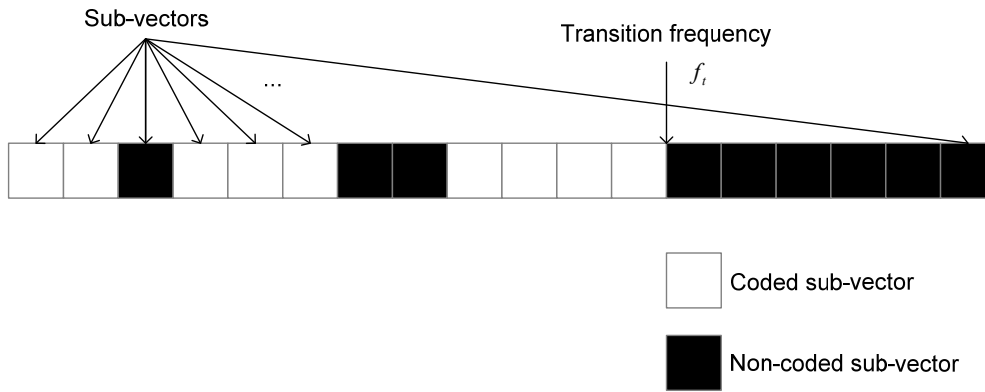


Figure 12/JT-G719 – Estimation of transition frequency  
(ITU-T G.719)

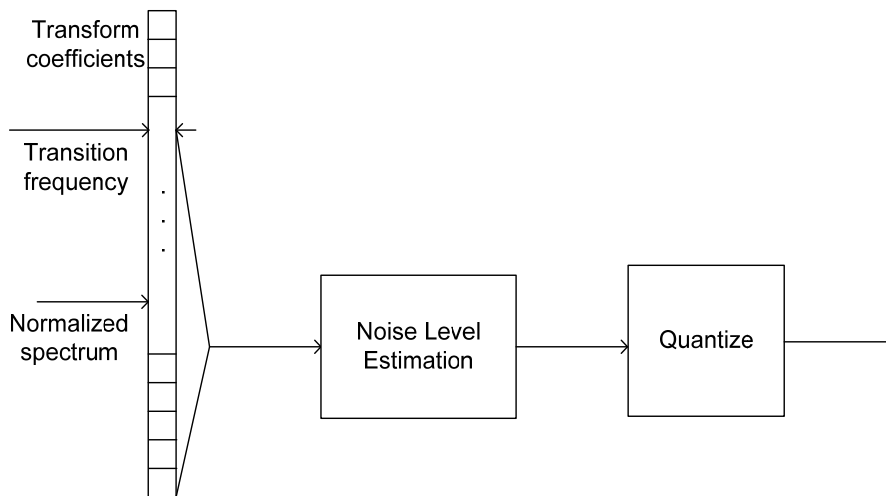


Figure 13/JT-G719 – Noise level estimation done below the transition frequency  
(ITU-T G.719)

次に雑音レベルは、遷移周波数  $f_t$  以下の正規化された、非符号化信号の平均レベルの測定によって評価される (Figure13/JT-G719 参照)。下式により雑音レベルが得られる

$$NoiseLevel = \frac{1}{|\mathfrak{S}|} \sum_{k \in \mathfrak{S}} \log_2(y(k)^2) - \log_2 \left\{ \frac{1}{|\mathfrak{S}|} \sum_{k \in \mathfrak{S}} y(k)^2 \right\} \quad (45)$$

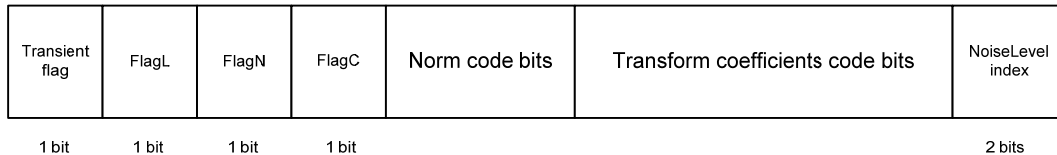
ここで、 $\mathfrak{S}$  は遷移周波数  $f_t$  以下で、1 ビットも割り当てられない係数インデックスの成す集合である。上式は、対数の凸性から常に 0 以下の値を返す。雑音レベルは、Table21/JT-G719 に従って 2 ビットのスカラ量子化器で量子化される。

Table 21/JT-G719 – Codebook entries for the uniform noiselevel quantizer  
(ITU-T G.719)

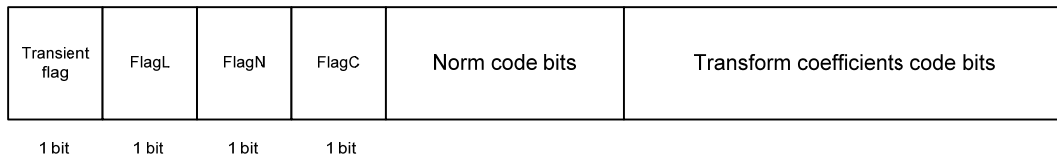
Index	Output Quantized NoiseLevel (dB)
0	0
1	-6
2	-12
3	-18

## 7. 8 ビット列形式

チャンネル上で伝送されるビット列は3つの共通部分からなる：*IsTransient*、*FlagL*、*FlagN*、*FlagC*の4つのフラグビットから成る補助情報、ノルムの符号ビット、そして変換係数の符号ビットである。定常状態の場合は、2ビットの雑音レベルインデックスも変換係数の符号ビットに続いて復号器に伝送される。Figure14 a)／JT-G719 と Figure14 b)／JT-G719 は、それぞれ定常状態と遷移状態におけるビット列の形式と伝送されるパラメータ領域の順序を図示している。



a) Case of stationary frames



b) Case of transient frames

Figure 14／JT-G719 – Bitstream format and transmission order  
(ITU-T G.719)

## 8 復号器の機能説明

### 8. 1 フラグの復号

ビット列復号の最初のステップは、種々のフラグの復号である。これらはビット列の最初の部分に位置している（7.8節参照）。

### 8. 2 ノルムの復号

*FlagN* の受信により、サブベクトルノルムの量子化インデックス  $I_N^q(p)$ 、 $p=0, \dots, P-1$  が符号ビットの復号によって復元され、ハフマン符号の復号に続いて差分復号が用いられるか、または直接、差分復号が行われる。実際の量子化ノルム  $N^q(p)$  はテーブル参照によって得られる（Table15／JT-G719）。

### 8. 3 スペクトル係数の復号

量子化されたノルムインデックス  $I_N^q(p)$ 、 $p=0, \dots, P-1$  を用いて、符号化器と同じ適応ビット割り当てがビット割り当てベクトル  $R(p)$ 、 $p=0, \dots, P-1$  を決定するために実行される。このステップは、7.5 節記載のステップと同様の手順で行われる。

スペクトル係数の復号は、*FlagC* の設定の有無に従って異なった方法で行われる。

*FlagC* が設定されていない場合は、ハフマン符号は用いらず、各サブベクトル  $p$  のインデックスはビット列から直接読みとられる。*FlagC* がセットされている場合、 $2 \leq R(p) \leq 5$  である各サブベクトル  $p$  に対しては、受信データはハフマン符号でありサブベクトルインデックスを得るために各サブベクトルに対する復号が必要である。 $2 \leq R(p) \leq 5$  でないときは、受信データは変換係数の量子化インデックスである。

レート 32 kbit/s においては、*FlagL* ビットがセットされていないときは、最初の3グループの係数 (I, II および III) に対する量子化インデックスのみが復元される。実際、Group IV に属するサブベクトルは量子化も伝送

もされない。

正規化された係数の逆量子化の手順は単純なもので、以下の通り実行される。

各サブベクトル  $p$  は、定常モードにおいては Table6,7,8,9/JT-G719 で、遷移モードにおいては Table11,12,13,14/JT-G719 で示すように、一つあるいは複数の 8 次元ベクトルから構成される。

各 8 次元ベクトル  $\mathbf{y}_{p,l}$  (ここで  $l$  は、サブベクトル  $p$  内の 8 次元ベクトルのインデックス) は以下の手順によって復号される：

- 1) 与えられたレート  $R(p)$  に応じたインデックスから符号ベクトル  $\mathbf{y}_{p,l}$  を見つける。
- 2) 符号ベクトル  $\mathbf{y}_{p,l}$  を、符号化器で用いられたスケールファクタ  $\beta$  または  $\beta(p)$  で再スケーリングする：

$$\bar{\mathbf{y}}_{p,l} = \mathbf{y}_{p,l} / \beta \quad \text{or} \quad \bar{\mathbf{y}}_{p,l} = \mathbf{y}_{p,l} / \beta(p) \quad (46)$$

- 3) 符号化手順のステップ 1) で用いたオフセット  $\mathbf{a}$  を、再スケーリングされた  $\bar{\mathbf{y}}_{p,l}$  に加える。

$$\tilde{\mathbf{y}}_{p,l} = \bar{\mathbf{y}}_{p,l} + \mathbf{a} \quad (47)$$

得られた 8 次元ベクトルは、復号された正規化スペクトル係数を表しており、 $y_N^q(k)$ ,  $k=0, \dots, 799$  に分類される。

符号化において、 $R(p)=0$  であるようなインデックス  $p$  のサブベクトルなど、ビットが割り当てられないサブベクトルが存在する場合がある。これらのサブベクトルは、符号化される係数を持たない。復号器において、これらの符号化されなかった係数は、8.4 節記載のスペクトル充填手法によって生成される。

#### 8.4 スペクトル充填

本章では、復号器にて適用されるスペクトル充填処理の技術概要を記述する。

受信したビット割り当てに基づき、遷移周波数  $f_t$  が、符号化器と同様な手順で推定される。スペクトル充填は、2つのアルゴリズムから構成されている。1つは、遷移周波数  $f_t$  までの低周波スペクトルの充填であり、もう1つは、低周波雑音充填スペクトルを用い、非符号化高周波要素を可能な限り再生成するものである。

これら2つのアルゴリズムの相互関係を、Figure 15/JT-G719 に示す。雑音充填アルゴリズムと帯域拡張の両方から得られるスペクトルは、受信した量子化ノルムによって形成された、正規化スペクトルである。

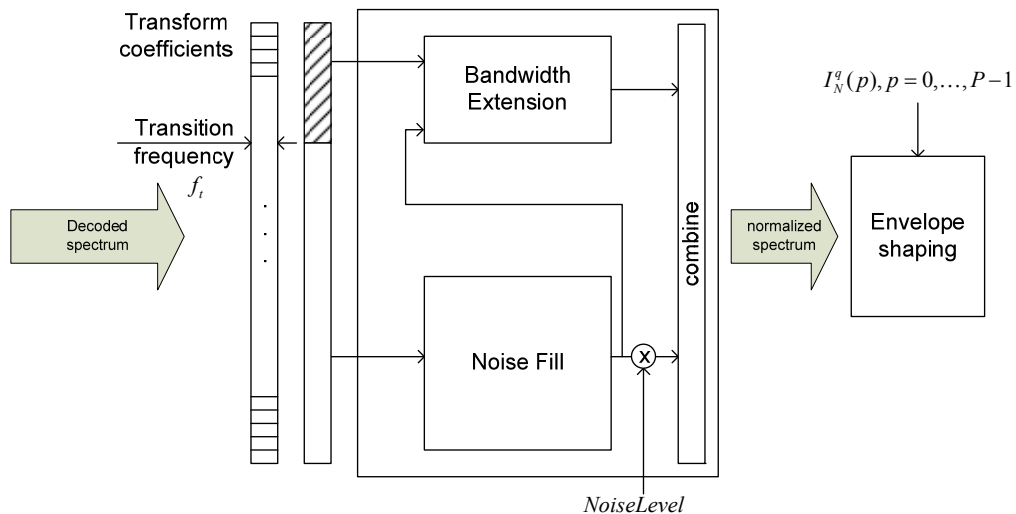


Figure 15/JT-G719 Spectrum Filling block diagram  
(ITU-T G.719)

### 8. 4. 1 雑音充填

雑音充填の最初のステップは、受信（復号）正規化変換係数から、いわゆるスペクトル符号帳の構築に依存している。このステップは、復号スペクトルの聴感的に関連づけられた係数を連結することで実現される。Figure 16/JT-G719 に、この手法を示す。復号スペクトルは、一定の長さのスペクトルホールと呼ばれる一連のゼロ係数を有している。この長さは、ビット割り当てのないサブベクトルの連続した長さの合計である。

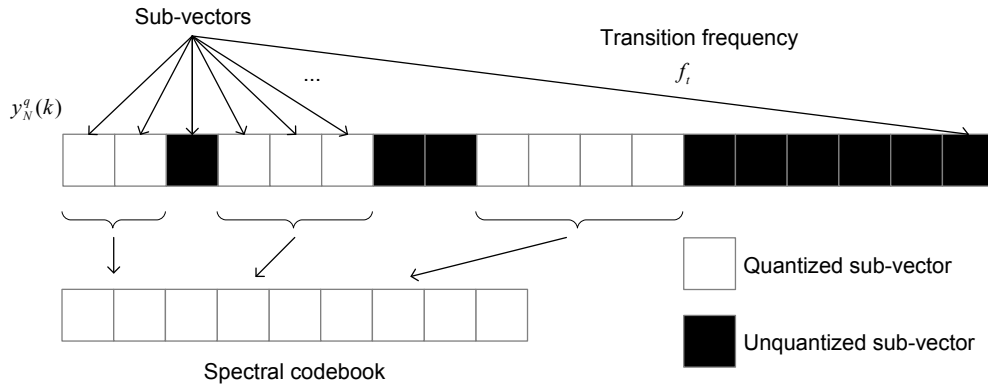


Figure 16/JT-G719 Building the spectral codebook from the decoded transform signal (ITU-T G.719)

全てのスペクトルホール長が、スペクトル符号帳の長さよりも長くなりうるため、符号帳要素は、いくつかのスペクトルホールを埋めるのに再利用される場合がある。

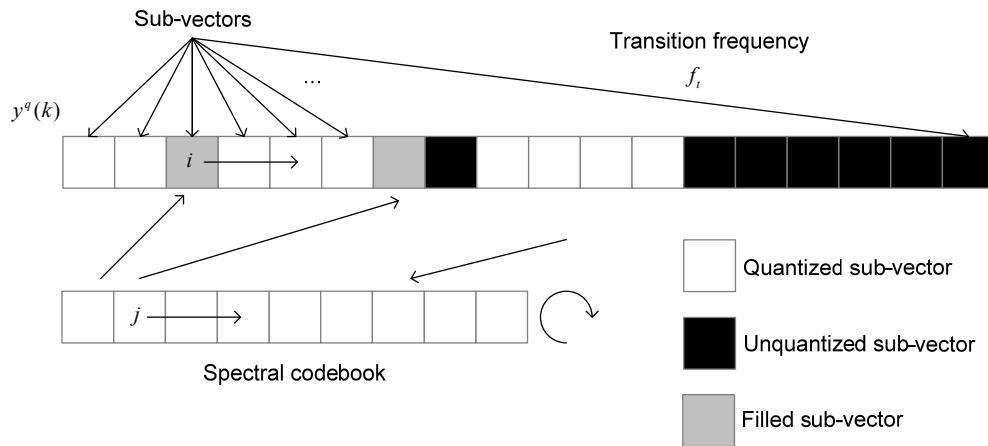


Figure 17/JT-G719 Noise filling from the spectral codebook up to the transition frequency (ITU-T G.719)

Figure 17/JT-G719 に、スペクトル符号帳 C に基づいて、量子化されていないスペクトル係数が、どのように埋められるかを示す。遷移周波数までの全てのスペクトルホールを網羅するために用いるインデックス  $i$  の増加と同様に、符号帳のインデックス  $j$  を増加することで、スペクトルホールが充填される。スペクトル符号帳の読みだしは、以下に示すように、巡回バッファとして連続的に実行される。

```

i = 0, j = 0
(1:) if R(pi) = 0 then yNq(i) = C(j),
      increment i, j (if j out of bound rewind j to start of codebook)
      if i = 0 then
          STOP
      else
          goto (1:)
      endif

```

(48)

#### 8. 4. 2 帯域拡張

雑音レベル減衰処理前の、前節で詳述した低周波数スペクトル充填に基づき、スペクトル充填の最終ステップで、フルバンドのオーディオ信号を生成する。言い換えると、その処理は、遷移周波数の値に基づいて、スペクトルを折り返して充填されたスペクトルから、高周波スペクトルを合成する。

フルバンドは、遷移周波数以下のスペクトル折り返しに基づいて、高周波スペクトル（遷移周波数以上のゼロ）を生成する。Figure 18/JT-G719 を参照のこと。最初のスペクトル折り返しは、遷移周波数  $f_t$  による対称点を境に実施される。 $f_t/2$  以下の周波数のスペクトル係数は、高周波へ折り返されることは無い。言い換えると、低周波数の上半分だけが折り返される。遷移周波数以上の全てのスペクトルを埋めるのに、低周波数の上半分の係数では足りない場合は、そのスペクトルは、最後に埋められた係数近辺が再度折り返される。

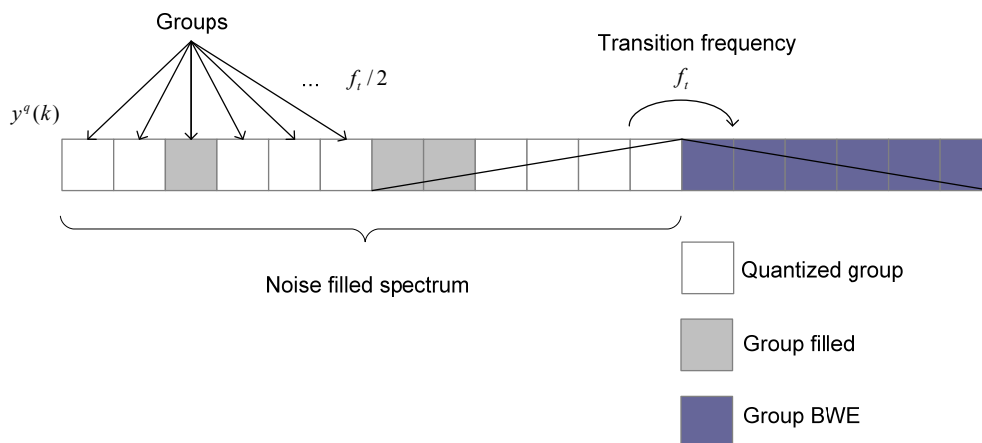


Figure 18/JT-G719 Full-bandwidth generation: the spectrum above the transition frequency is regenerated using spectral folding from the transition frequency (ITU-T G.719)

この処理は、フルバンド信号が生成されるまで繰り返される。実際のところ、フルバンドに到達するのに、最大2回の繰り返して十分である。

#### 8. 4. 3 雑音レベル調整

スペクトルホールの微細構造が決定された後、スペクトルの雑音充填部分が、受信 *NoiseLevel* インデックスによって減衰される。過渡モードの場合、その *NoiseLevel* は、符号化器では用いられず（評価されず）、ゼロインデックス、すなわち 0 dB に自動的に設定される。

この処理を、下式にまとめる。

$$y_N^q(k) = 2^{-\text{NoiseLevel}} y_N^q(k), \text{ for } k \text{ such that } R(p_k) = 0 \text{ and } k \leq f_t \quad (49)$$

#### 8. 4. 4 スペクトル充填包絡シェーピング

フルバンド微細スペクトル構造が生成された時、スペクトルは、下式の受信量子化サブベクトルノルムによってシェーピングされる。

$$z^q(k) = 2^{I_k^q(p_k)} y_N^q(k), k = 0, \dots, 799 \quad (50)$$

ここで、 $p_k$  は、スペクトル係数  $k$  (Table 6-9/JT-G719 が定常モード、Table 11-14/JT-G719 が過渡モード) に対応するサブベクトルインデックスである。周波数 20kHz から 24kHz に対応する 799 から 959 までの係数は、以下のようにゼロに設定される。

$$z^q(k) = 0, k = 800, \dots, 959 \quad (51)$$

#### 8. 5 逆変換

逆変換は、符号化器と同じ原理に基づいている。Figure 19/JT-G719 に、それぞれの復号スペクトル  $z_{(r)}^q(k)$  に対するタイムチャートを示す。ここで、 $r$  はフレームインデックスである。それぞれのフレーム  $r$  に使用される逆変換の型は、受信フラグ  $IsTransient^{(r)}$  によって決められる。

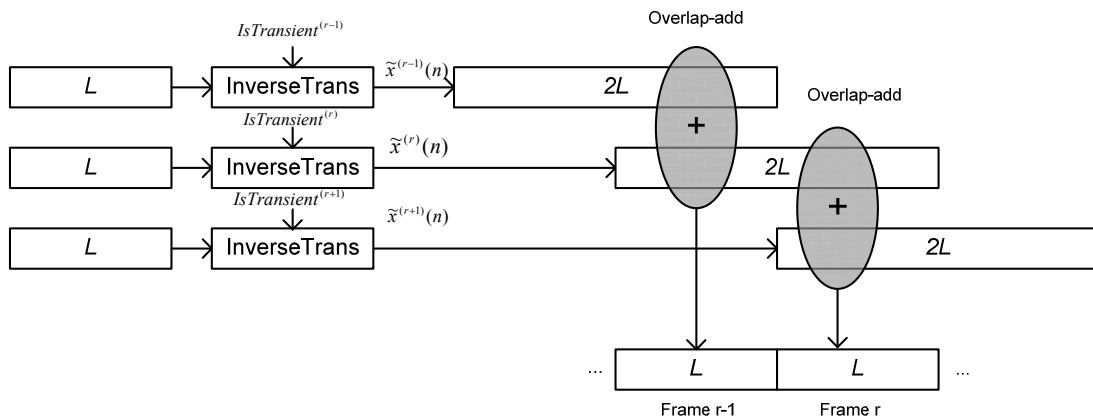


Figure 19/JT-G719 Time chart of signal buffering and inverse transform  
(ITU-T G.719)

符号化器と同様、過渡モードと定常モードは、同一のバッファを使用する。そのスペクトル、適応時間周波数変換によって最初の逆変換が実施される。そのフレームは 40ms 信号である。時間軸信号に復元するため、これらのフレームは、窓かけ処理され、重ね合わせ加算される。この処理を、Figure 19/JT-G719 に示す。それぞれの入力ビット列フレームに対して、復号器は、前フレームに対する重ね合わせ加算処理のためのバッファを必要とする。

##### 8. 5. 1 信号の事前正規化と事後正規化

これらの処理は、逆変換計算の前後に適用され、符号化器と同一の処理である。7.2.3 及び 7.2.7 節を参照のこと。そのスペクトルは、32 ビット符号付き整数の全レンジが全て使われるように正規化され、逆変換の出力は、逆正規化される。

### 8. 5. 2 定常モード

*IsTransient* フラグが *FALSE* の時、IV 型逆離散コサイン変換 (IDCT<sub>IV</sub>) は、時間エイリアシング信号  $\tilde{x}(n)$ ,  $n=0, \dots, L-1$  を導き出すスペクトルに適用される。

その IDCT<sub>IV</sub> は、DCT<sub>IV</sub> と同一であり、下記式により与えられる。ここで、正規化処理は省略してある。

$$\tilde{x}^q(n) = \sum_{k=0}^{L-1} z^q(k) \cos\left[\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\frac{\pi}{L}\right], \quad n=0, \dots, L-1 \quad (52)$$

### 8. 5. 3 過渡モード

*IsTransient* フラグが *TRUE* の時、スペクトルは、始めに4つのスペクトル  $z_m^q(k)$ ,  $m=0, \dots, 3$ , に逆インターリーブされる。この演算 (処理) は、符号化器におけるインターリーブ処理の逆であり、単純なバッファへのコピーと再インデックス処理で構成されている。

短時間 5 ms 変換に対応するこの4つのスペクトルは、短時間逆 DCT<sub>IV</sub> 変換を用い時間エイリアシング領域へ始めに変換される。得られた信号は、 $\tilde{x}_m^{qw}(n)$ ,  $m=0, \dots, 3$  と記述し、それぞれの長さは  $L/4$  である。

得られた時間域エイリアシング信号は、逆時間域エイリアシング処理を用い、時間領域においてさらに拡張される。この処理は、(8) 式の疑似逆行列として表現される ( $L$  を  $L/4$  で置き換える)。

正確には、下式によって演算される。

$$\tilde{\mathbf{x}}_m^{qw} = \begin{bmatrix} 0 & \mathbf{I}_{L/8} \\ 0 & -\mathbf{J}_{L/8} \\ -\mathbf{J}_{L/8} & 0 \\ -\mathbf{I}_{L/8} & 0 \end{bmatrix} \tilde{\mathbf{x}}_m^q \quad (53)$$

それぞれのサブフレームのインデックス  $m$  に対して得られる信号の長さは、入力スペクトルの長さの倍に等しく、 $L/2$  である。

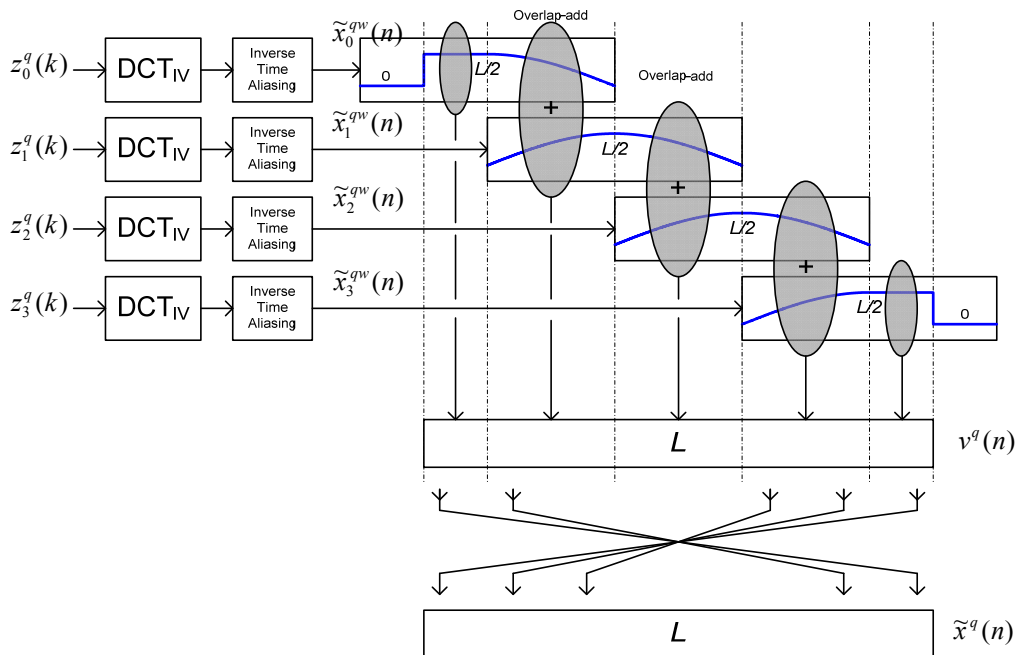


Figure 20/JT-G719 Algorithm for inverse transform in the case of Transient Mode (ITU-T G.719)



それぞれのサブフレーム  $l$  に対して得られる時間域エリアシング信号は、符号化器と同窓形状にて窓処理される。得られた窓処理済み信号は、重ね合わせ加算される。ここで、最初の  $m=0$  と最後の  $m=3$  のサブフレームの窓は、ゼロである。これは、符号化器において、ゼロ埋め処理されているからである。これら2つのフレームエッジは、演算される必要があり、実質的には落とされる。全てのサブフレーム  $v^q(n)$  の重ね合わせ加算の結果得られる信号は、符号化器における処理の逆演算により再整列される。信号  $\tilde{x}^q(n)$ ,  $n=0, \dots, L-1$  を導き出す (13) 式、及び、Figure 20/JT-G719 を参照のこと。これらの処理の概要は、Figure 20/JT-G719 に示されている。

#### 8. 5. 4 窓処理と重ね合わせ加算

定常モード、あるいは、過渡モードにおける逆変換出力は、 $L$  の長さの出力である。窓処理の前の信号は、下式による  $2L$  長の逆時間エリアシング信号となる。

$$\tilde{\mathbf{x}}^{wq} = \begin{bmatrix} 0 & \mathbf{I}_{L/2} \\ 0 & -\mathbf{J}_{L/2} \\ -\mathbf{J}_{L/2} & 0 \\ -\mathbf{I}_{L/2} & 0 \end{bmatrix} \tilde{\mathbf{x}}^q \quad (54)$$

得られた信号は、(6) 式のサイン窓を用いて、それぞれのフレーム  $r$  に対して窓処理される。

$$\tilde{x}^{(r)}(n) = h(n)\tilde{x}_{(r)}^{wq}(n), \quad n=0, \dots, 2L-1 \quad (55)$$

最後に、出力フルバンド信号は、2つの連続するフレームに対して信号  $\tilde{x}^{(r)}(n)$  を重ね合わせ加算することで再構築される。

$$x^{(r)}(n) = \tilde{x}^{(r-1)}(n+L) + \tilde{x}^{(r)}(n), \quad n=0, \dots, 2L-1 \quad (56)$$

#### 8. 6 フレーム消失補償

フレーム消失補償機能は、復号器に組み込まれている。フレームが正しく受信された時は、再生変換係数がバッファに格納される。復号器が、フレーム消失、あるいは、破損を検出した時は、最後に受信したフレームの再生変換係数は、0.5 倍で減衰処理され、現フレームに対する再生変換係数として使用される。この処理は、これらを時間領域に変換し、窓処理重ね合わせ加算演算することにより実現される。

### 9 柔軟なレート選択

7章、8章にて詳述したコーデックアルゴリズムは、各動作に対して規定された総ビット割当量のみ柔軟的である。生成されるビット列は、自己解凍可能であり、そのビット列を正しく復号するためにアウトバンドでシグナリングされる総ビット数を必要とする。

通常の 32、48、そして 64kbit/s レートに加え、コーデックは、柔軟なレート選択が可能である。実際のところ、32kbit/s から 64kbit/s までの間、4kbit/s ステップでいかなるレートも適用可能である。さらに、このコーデックは、64kbit/s から 128kbit/s までの高レートも提供できる。コーデックの動作モードをシグナリングにより適用させるため、Table 22/JT-G719 にあらかじめ定められたレートセットを示す。

Table 22/JT-G719 Predefined set of rates  
(ITU-T G.719)

Mode	Frame size (bytes)	Rate (kbit/s)	Mode	Frame size (bytes)	Rate (kbit/s)
0	80	32	10	180	72
1	90	36	11	190	76
2	100	40	12	200	80
3	110	44	13	210	84
4	120	48	14	220	88
5	130	52	15	240	96
6	140	56	16	260	104
7	150	60	17	280	112
8	160	64	18	300	120
9	170	68	19	320	128

## 10 シミュレーションソフトウェア

JT-G719コーデックは、ITU-T Rec. G.191 Software Tool Library(2005)に規定される基本演算子バージョン2.2を用い、固定小数点で実行可能である。

16/32ビットのJT-G719コーデックをシミュレートするANSI-Cコードは、ITU-TのWebサイトから入手可能である。以下の節では、このシミュレーションコードの使用方法、およびそのソフトウェアがどのように構成されているかを概説している。

### 10.1 シミュレーションソフトウェアの使用方法

JT-G719コーデック符号化器のコマンドラインは、以下の通りである。

```
encoder -r R -i input.raw -o bitstream_file.g192
```

where  
R is the desired bitrate in bits/s;  
input.raw is the input 16-bit linear PCM file sampled at 48 kHz  
bitstream\_file.g192 is the desired G.192 bitstream file

JT-G719コーデック復号器のコマンドラインは、以下の通りである。

```
decoder -i bitstream_file.g192 -o output.raw
```

where  
bitstream\_file.g192 is the input G.192 bitstream file  
output.raw is the output 16-bit linear PCM file sampled at 48 kHz

### 10.2 シミュレーションソフトウェアの構成

シミュレーションソフトウェアは、大きく3つの部分から構成される。すなわち、符号化器ソースコード、復号器ソースコード、そして、共通部分ソースコードである。Table 23/JT-G719に、これら3つのソースコードのコーデックで用いられるROMテーブルを示す。

符号化器と復号器のコードは、各個別ルーチンと共通ルーチンとに分割されている。符号化器用ルーチンは、Table 24/JT-G719に示され、復号器用ルーチンは、Table 25/JT-G719に示される。共通用ルーチンは、Table 26/JT-G719に示される。

Table 23/JT-G719 Tables in C code  
(ITU-T G.719)

Table name	Size (Words)	Description
A	20	Table for calculating quantization weights
cos_win	480	Cosine Table for transform
dct480_table_1	2*480	Table used for 960-point DCT <sub>IV</sub>
dct480_table_2	2*480	Table used for 960-point DCT <sub>IV</sub>
dct120_table_1	2*120	Table used for 240-point DCT <sub>IV</sub>
dct120_table_2	2*120	Table used for 240-point DCT <sub>IV</sub>
dic0	80	Codebook number 0 of LVQ1
dic1	80	Codebook number 1 of LVQ1
dic1Q8	80	Codebook number 1 dic1 in Q8
dic2	10	Codebook number 2 of LVQ1
dic3	256	Codebook number 3 of LVQ1
dic4	2048	Codebook number 4 of LVQ1
dicnlg2	40	Codebook for norm quantization
FacLVQ2HalfQv	10	Table used for LVQ2 code-vector searching
FacLVQ2Mask	10	Table used for LVQ2 code-vector searching
FacLVQ2Qv	10	Table used for LVQ2 code-vector searching
fft120cnst	144	Table used for Winograd 120-point FFT
huffcoef	60	Huffman coding Table for coefficient indices
huffnorm	32	Huffman coding Table for norm indices
huffoffset	6	Huffman coding Table for coefficient indices
huffsize	60	Huffman coding Table for coefficient indices
huffsizn	32	Huffman coding Table for norm indices
indxPost	2*120	Table for Winograd 120-point FFT
indxPre	2*120	Table for Winograd 120-point FFT
inv_sbfsiz	4	Table used for norm quantization
moreshift	4	Table used for norm quantization
ptwdf	2*480	Table used for Winograd 480-point FFT
RV	10	Table for quantizing transform coefficients
sfm_end	44	Table used for spectrum filling (ending band)
sfm_start	44	Table used for spectrum filling (starting band)
sfm_width	20	Table for calculating quantization weights
sfmsize	44	Table used for bit allocation
short_window	480	Window for 5-ms transform
sin_win	480	Sine Table for transform
window	1920	Window for 20-ms transform
wscw16q15	240	Table used for MDCT

Table 24/JT-G719 Summary of encoder specific routines  
(ITU-T G.719)

Filename	Description
bitallocsum.c	Calculate the total number of bits allocated over frame
code2idx.c	Finding the index of an LVQ2 code-vector
detect_transient.c	Transient detection
diffcod.c	Differential coding for indices of quantized norms
encode_frame.c	Main encoding - encode a single frame
encoder.c	Encoder wrapper
encode_int.c	Initialization of encoder
encoder_rom.c	Tables used for encoding
flvqenc.c	Quantization and coding of transform coefficients
huffcheck.c	Check Huffman encoding for LVQ2 indices
logqnorm.c	Log2 quantization for norms of sub-vectors
lvq1.c	$RE_{\delta}$ -based lattice vector quantization (LVQ1)
lvq2.c	$D_{\delta}$ -based lattice vector quantization (LVQ2)
noise_adjust.c	Calculate Noiselevel attenuation
normalizecoefs.c	Normalization of transform coefficients with quantized norms
packingc.c	Huffman coding and packing indices of quantized coefficients
procnf.c	Quantization for sub-vectors originally allocated with 0 bits
procnobitsbfm.c	Encoding sub-vectors originally allocated with 0 bits
qcoefs.c	Fast Lattice Vector Quantization (FLVQ) for normalized coefficients
reordernorm.c	Reorder quantized norms and their indices for Transient Mode
trans_direct.c	Transformation of signal from time domain to DCT domain
wtda.c	Windowing and Time Domain Aliasing (TDA)

Table 25/JT-G719 Summary of decoder specific routines  
(ITU-T G.719)

Filename	Description
decode_frame.c	Main decoding - decode a single frame
decoder.c	Decoder wrapper
decode_init.c	Initialization of decoder
decoder_rom.c	Tables used for decoding
dprocnf.c	De-quantization for sub-vectors originally allocated with 0 bits
dprocnobitsbfm.c	Decoding sub-vectors originally allocated with 0 bits
dqcoefs.c	Decoding of Fast Lattice Vector Quantization (FLVQ)
fill_spectrum.c	Fill the spectrum which has been quantized with 0 bits
flvqdec.c	Decoding and de-quantization of transform coefficients
hdeclvq.c	Huffman decoding for indices of quantized transform coefficients
hdecnrm.c	Huffman decoding for indices of quantized norms
trans_inv.c	Transformation of signal from DCT domain to time domain
unpackc.c	Huffman decoding and unpacking indices for quantized coefficients
window_ola.c	Windowing, overlap, and add

Table 26/JT-G719 Summary of common routines and headers  
(ITU-T G.719)

<b>Filename</b>	<b>Description</b>
bitalloc.c	Adaptive bit-allocation based on quantized norms
bitstream.c	Conversion between coding indices and ITU G.192 bit stream
codesearch.c	Finding the closest point of lattice to an arbitrary vector
common_rom.c	Tables used for both encoding and decoding
complxop.c	Complex basic operations
det.c	Fast DCT-IV algorithm using FFT
idx2code.c	Finding an LVQ2 code-vector from its index
Interleave_spectrum.c	Interleaving and de-interleaving spectrum for Transient Mode
recovernorm.c	Recover reordered norms and their indices for Transient Mode
reordvct.c	Rearrange a vector in decreasing order
weight.c	Calculate quantization weights
cnst.h	Definition of constants
complxop.h	Function prototypes for complex basic operations
proto.h	Function prototypes for codec
rom.h	Definition of tables
state.h	Definition of state structures of codec

付属資料A  
(標準 J T - G 7 1 9 に対する)

蓄積形式の定義

A. 1 適用範囲

本付属資料は、圧縮した標準 J T - G 7 1 9 フレームの蓄積形式を定義する。サンプルエントリの定義は、標準 J T - G 7 1 9 フレームのデフォルトの蓄積形式として、ISO ベースメディアファイル形式の登録ポイントを提供する。標準 J T - G 7 1 9 コンテンツを保存するために、ISO ベースメディアファイル形式から派生する全ファイル形式は、この定義を使用しなければならない。

A. 2 必須文献

下記の T T C 標準、I T U - T 勧告、及び他の参考文献は、この本文の参考文献を通して、本標準の規定を構成するものである。出版された時点でその版が適用され、全ての標準および他の参照文献は、改定に従うものとする。従って、本標準のユーザには、以下の T T C 標準やその他の参考すべき文献について、最新の版が適用の可能性を調査するよう奨励される。現在、最新版の T T C 標準、及び I T U - T 勧告のリストは、定期的に出版されている。

本標準内での文章の参照は、単独の文書としては、それを標準の扱いとはしない。

[1] ISO/IEC 14496-12:2005 | 15444-12:2005

Information technology – Coding of audio-visual objects – Part 12: ISO base media file format | Information technology – JPEG 2000 image coding system – Part 12: ISO base media file format

A. 3 参考文献

[2] 3GPP TS 26.244

Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)

[3] ISO/IEC 14496-14:2003

Information technology – Coding of audio-visual objects – Part 14: MP4 file format

[4] Apple Computer

Audio Interchange File Format AIFF-C, August 1991 (also at ftp://ftp.sgi.com/sgi/aiff-c.9.26.91.ps.Z)

A. 4 標準 J T - G 7 1 9 ビット列の蓄積形式

A. 4. 1 単一チャンネル蓄積形式

本節では、標準 J T - G 7 1 9 圧縮フレームを保存する方法について規定する。圧縮フレームは、標準 J T - G 7 1 9 の 7. 8 節に定義される。標準 J T - G 7 1 9 のビット列は、8 ビットに分割され、最も左のビット（最上位ビット）から、右端ビット（最下位ビット）の順序で伝送される。プレフィックスバイト、もしくは、コンテンツバイトテーブル(ToC)が、圧縮オーディオフレームデータの先頭に付加される。プレフィックスバイトは、下記の表で与えられる。Table A.1/JT-G719 に、各領域の意味を示す。

Table A.1/JT-G719 Format of the prefix byte (ToC)  
(ITU-T G.719)

Bit	0	1	2	3	4	5	6	7	
Name	F	L					R		R
Default value	0						0	0	

F (1ビット) : 常にゼロに設定する。

L (5ビット) : 個々のフレームのフレーム長を与える領域である。領域値とフレーム長の関係を、Table A.2/JT-G719 に示す。

Table A.2/JT-G719 Relationship between field value and frame length  
(ITU-T G.719)

L	Length (bytes)
0	0 NO_DATA
1-7	N/A (reserved)
8-22	$80+10*(L-8)$
23-27	$240+20*(L-23)$
28-31	N/A (reserved)

L = 0 の値は、空フレーム (0 バイト) を示すために使用される。これは、例えば生の会話録音時にフレームが消失する場合などに有効である。[1..7] と [28..31] の範囲にある値は予約されており、1 フレームの ToC の中にこれらの値が存在する場合は、フレーム全体が無効なフレームとして扱われ廃棄される。

幾つかの例を Table A.3/JT-G719 に示す。ここで、フレームサイズと対応する符号化ビットレートは、L の値に基づいて算出される。

Table A.3/JT-G719 Examples of L values and corresponding frame lengths  
(ITU-T G.719)

L	Bytes	Bitrate (kbit/s)
8	80	32
9	90	36
10	100	40
12	120	48
16	160	64
22	220	88
23	240	96
25	280	112
27	320	128

この符号化方法では、32~128kbit/s の定義範囲において、32~88kbit/s では 4kbit/s のステップ、88~128kbit/s では 8kbps のステップがそれぞれ実現できる。

R (2ビット) : 予約ビットである。送信時は 0 が設定され、受信時は無視される。

#### A. 4. 2 複数チャンネルオーディオのサポート

標準 JT-G 7 1 9 符号器自体は、単一ビット列に含まれる複数チャンネルのオーディオコンテンツ符号化をサポートしないが、それぞれ個々のチャンネルを個別に符号化/復号するために使用することができる。個別に符号化された複数のチャンネルコンテンツを保存するために、同一の 20ms 区間にフレーム化、符号化した全チャンネルのオーディオフレームを 1 つのフレームブロックに論理的に収集する。フレームブロックにおいてそれぞれに圧縮されたフレームには、固有の T o C プレフィックスバイトが付加される。

セッションのチャンネル数を示すためには、帯域外のシグナリングを使用する必要がある。各フレームブロック内において、異なるチャンネルのオーディオフレームの順序は、A I F F - C オーディオインターチェンジ形式[4]に示された規定に従う。

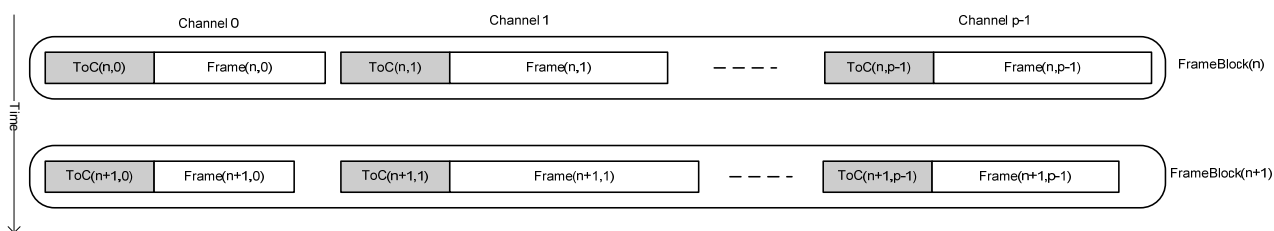


Figure A.1/JT-G719 Collection of multi-channel frames in frame blocks  
(ITU-T G.719)

20ms フレームを論理的に収集するには、Figure A.1/JT-G719 に示すように、異なるチャンネルを単純に交互に処理すればよい。

#### A. 5 ISOベースファイル形式のサンプルエントリ記述

ISOベースメディアファイル形式[1]に関するファイルにおいて、サンプル記述ボックスには、使用する符号化種類の情報、及び符号化に必要な全ての初期化情報が与えられる。サンプル記述ボックスは、Figure A.2/JT-G719 に示されるISOのファイル形式のボックス階層構造に見ることができる。

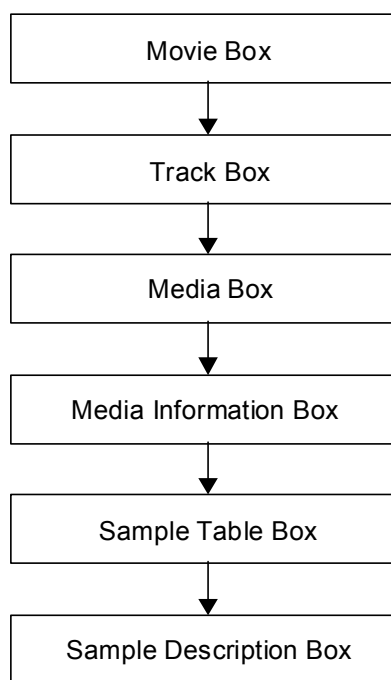


Figure A.2/JT-G719 ISO File Format Box Structure Hierarchy  
(ITU-T G.719)

サンプル記述ボックスは、1つまたは複数のサンプルエントリを持つことができる。有効なサンプルエントリの例としては、AMRSampleEntry、MP4AudioSampleEntry、及びMP4VisualSampleEntryを含む3GP(3GPP TS 26.244)、及びMP4(ISO/IEC 14496-14:2003)ファイルが既に定義されている。なお、標準JT-G719のサンプルエントリとしては、G719SampleEntryが定義される。



A. 5. 1 サンプルエントリボックス

G719SampleEntry ボックスのボックス種類は、g719 である。

G719SampleEntry ボックスを、以下に定義する (Table A.4/JT-G719 も合わせて参照すること)。

G719SampleEntry ::= BoxHeader

Reserved\_6  
 Data-reference-index  
 Reserved\_8  
 NumberOfChannels  
 Reserved\_2  
 Reserved\_4  
 SamplingRate  
 Reserved\_2  
 G719SpecificBox

Table A.4/JT-G719 G719SampleEntry fields  
 (ITU-T G.719)

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'g719'
Reserved_6	Unsigned int(8) [6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference boxes.	
Reserved_8	Const unsigned int(32) [2]		0
NumberOfChannels	Unsigned int(16)	Set to the number of channels, e.g. 1 (mono), 2 (stereo), etc.	
Reserved_2	Const unsigned int(16)		16
Reserved_4	Const unsigned int(32)		0
SamplingRate	Const Unsigned int(16)	Audio programme sampling rate	48000
Reserved_2	Const unsigned int(16)		0
G719SpecificBox		Information specific to the G.719 decoder.	

A. 5. 2 G719SampleEntryボックスに対するG719SpecificBox領域

標準 J T - G 7 1 9 の G719SpecificBox 領域を、Table A.5/JT-G719 に定義する。ファイルに標準 J T - G 7 1 9 圧縮メディア情報が含まれる場合、G719SampleEntry ボックスに対して G719SpecificBox が常に含まれなければならない。

Table A.5/JT-G719 The G719SpecificBox fields for G719SampleEntry  
 (ITU-T G.719)

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'd719'
DecSpecificInfo	G719DecSpecStruc	Structure which holds the G.719 Specific information	

BoxHeader Size and Type : 標準 J T - G 7 1 9 復号器の仕様ボックスのサイズと種類を示す。種類は d719 で

なければならない。

DecSpecificInfo：標準 J T - G 7 1 9 ストリームの仕様情報を登録する構造体である。

構造体 G719DecSpecStruc の定義を以下に示す。

```
struct G719DecSpecStruc{
    Unsigned int (32)      vendor
    Unsigned int (8)      decoder_version
}
```

G719DecSpecStruc メンバの定義を、以下に示す。

**vendor**：符号器の製造元を示す 4 文字コードである（例えば、VXYZ）。ベンダ領域は、符号化データを生成するために使用した符号器に関するベンダ情報を提供する。これは、復号終了までに使用される重要な領域である。製造元が既に 4 文字コードを取得している場合は、この領域に同様のコードを使用することが推奨されている。また、製造元が、製造元の名前を簡潔に表す 4 文字コードを生成することも推奨されている。なお、上記を無視しても問題はない。

**decoder\_version**：ベンダ復号器のバージョンを示す。ベンダの復号器は、最良の方法(例えば、最適な方法)で、符号化したストリームを復号することができる。この領域は、ベンダ領域に密接に関係する。例えば、最適化された符号器－復号器のバージョンの組み合わせをもつベンダには有効である。ベンダにとって復号器のバージョンが重要ではない場合、設定値をゼロにする。また、上記を無視しても問題はない。

付属資料B  
(標準 J T - G 7 1 9 に対する)  
**浮動小数点実装**

**B. 1 適用範囲**

本付属資料は、標準 J T - G 7 1 9 符号化アルゴリズムの浮動小数点実装について記述している。

**B. 2 アルゴリズム記述**

標準 J T - G 7 1 9 の浮動小数点版は、固定小数点版と同じアルゴリズムステップを踏んでいる。同様に、ビット列形式は、標準 J T - G 7 1 9 の固定小数点版と同一である。符号化アルゴリズムの完全な記述については、標準 J T - G 7 1 9 の本体を参照すること。

**B. 3 ANS I Cコード**

浮動小数点版の標準 J T - G 7 1 9 符号器をシミュレーションする ANS I Cコードは、ITU-T の Web サイトから入手が可能である。以下に、シミュレーションコードの使用方法和、ソフトウェアの構成についてまとめる。

**B. 3. 1 シミュレーションソフトウェアの使用法**

標準 J T - G 7 1 9 符号器のコマンドラインを以下に示す。

```
encoder -r R -i input.raw -o bitstream_file.g192
```

ここで、

R は、bit/s 単位での指定ビットレートを示す。

input.raw は、48kHz サンプリング、16bit リニア PCM ファイル入力である。

bitstream\_file.g192 は、指定する標準 J T - G 1 9 2 ビット列ファイル名を示す。

標準 J T - G 7 1 9 復号器のコマンドラインを以下に示す。

```
decoder -i bitstream_file.g192 -o output.raw
```

ここで、

bitstream\_file.g192 は、標準 J T - G 1 9 2 ビット列ファイル入力を示す。

output.raw は、48kHz サンプリング、16bit リニア PCM ファイル出力である。

**B. 3. 2 シミュレーションソフトウェアの構成**

シミュレーション用のソフトウェアは、3つの主要な部分から構成される。符号器のソースコード、復号器のソースコード、共通ソースコードである。Table B.1/JT-G719 は、3つの主要な部分のソースコードにおいて、符号器が使用する ROM テーブルの一覧である。符号器と復号器のコードは、特定ルーチンと共通ルーチンから構成される。特定の符号器のルーチンを Table B.2/JT-G719 に、特定の復号器のルーチンを Table B.3/JT-G719 に示す。また、共通ルーチンを Table B.4/JT-G719 に示す。

Table B.1/JT-G719 Tables in C code  
(ITU-T G.719)

Table name	Size (Words)	Description
A	20	Table for calculating quantization weights
dct480_table_1	2*960	Table used for 960-point DCTIV
dct480_table_2	2*960	Table used for 960-point DCTIV

Table name	Size (Words)	Description
dct120_table_1	2*240	Table used for 240-point DCTIV
dct120_table_2	2*240	Table used for 240-point DCTIV
Dic0	80	Codebook number 0 of LVQ1
Dic1	80	Codebook number 1 of LVQ1
Dic2	10	Codebook number 2 of LVQ1
Dic3	256	Codebook number 3 of LVQ1
Dic4	2048	Codebook number 4 of LVQ1
Dicn	80	Codebook for quantization of norms
dicnlg2	40	Codebook for norm quantization
FacLVQ2HalfQv	10	Table used for LVQ2 code-vector searching
FacLVQ2Mask	10	Table used for LVQ2 code-vector searching
FacLVQ2Qv	10	Table used for LVQ2 code-vector searching
Fft120cnst	288	Table used for Winograd 120-point FFT
Huffcoef	60	Huffman coding Table for coefficient indices
Huffnorm	32	Huffman coding Table for norm indices
Huffoffset	6	Huffman coding Table for coefficient indices
Huffsize	60	Huffman coding Table for coefficient indices
Huffsize	32	Huffman coding Table for norm indices
indxPost	240	Table for Winograd 120-point FFT
indxPre	240	Table for Winograd 120-point FFT
Ptwdf	2*960	Table used for Winograd 480-point FFT
RV	10	Table for quantizing transform coefficients
sfm_end	44	Table used for spectrum filling (ending band)
sfm_start	44	Table used for spectrum filling (starting band)
sfm_width	20	Table for calculating quantization weights
Sfmsize	44	Table used for bit allocation
short_window	960	Window for 5-ms transform
Thren	78	Searching thresholds for quantization of norms
Window	3840	Window for 20-ms transform
wscw16q15	480	Table used for MDCT

Table B.2/JT-G719 Summary of encoder specific routines  
(ITU-T G.719)

Filename	Description
bitallocsum.c	Calculate the total number of bits allocated over frame
Code2idx.c	Finding the index of an LVQ2 code-vector
detect_transient.c	Transient detection
diffcod.c	Differential coding for indices of quantized norms
encode_frame.c	Main encoding - encode a single frame
encoder.c	Encoder wrapper
encode_int.c	Initialization of encoder
encoder_rom.c	Tables used for encoding
flvqenc.c	Quantization and coding of transform coefficients
huffcheck.c	Check Huffman encoding for LVQ2 indices
logqnorm.c	Log2 quantization for norms of sub-vectors
lvq1.c	RE8-based lattice vector quantization (LVQ1)

Filename	Description
lvq2.c	D8-based lattice vector quantization (LVQ2)
noise_adjust.c	Calculate Noiselevel attenuation
normalizecoefs.c	Normalization of transform coefficients with quantized norms
packing.c	Huffman coding and packing indices of quantized coefficients
procnf.c	Quantization for sub-vectors originally allocated with 0 bits
procnobitsbfm.c	Encoding sub-vectors originally allocated with 0 bits
qcoefs.c	Fast Lattice Vector Quantization (FLVQ) for normalized coefficients
reordernorm.c	Reorder quantized norms and their indices for Transient Mode
trans_direct.c	Transformation of signal from time domain to DCT domain
wtda.c	Windowing and Time Domain Aliasing (TDA)

Table B.3/JT-G719 Summary of decoder specific routines  
(ITU-T G.719)

Filename	Description
decode_frame.c	Main decoding - decode a single frame
decoder.c	Decoder wrapper
decode_init.c	Initialization of decoder
decoder_rom.c	Tables used for decoding
dprocnf.c	De-quantization for sub-vectors originally allocated with 0 bits
dprocnobitsbfm.c	Decoding sub-vectors originally allocated with 0 bits
dqcoefs.c	Decoding of Fast Lattice Vector Quantization (FLVQ)
fill_spectrum.c	Fill the spectrum which has been quantized with 0 bits
flvqdec.c	Decoding and de-quantization of transform coefficients
hdeclvq.c	Huffman decoding for indices of quantized transform coefficients
hdecnrm.c	Huffman decoding for indices of quantized norms
trans_inv.c	Transformation of signal from DCT domain to time domain
unpackc.c	Huffman decoding and unpacking indices for quantized coefficients
window_ola.c	Windowing, overlap, and add

Table B.4/JT-G719 Summary of common routines and headers  
(ITU-T G.719)

Filename	Description
bitalloc.c	Adaptive bit-allocation based on quantized norms
bitstream.c	Conversion between coding indices and ITU G.192 bit stream
codesearch.c	Finding the closest point of lattice to an arbitrary vector
common_rom.c	Tables used for both encoding and decoding
complxop.c	Complex basic operations
dct.c	Fast DCT-IV algorithm using FFT
idx2code.c	Finding an LVQ2 code-vector from its index
Interleave_spectrum.c	Interleaving and de-interleaving spectrum for Transient Mode
recovernorm.c	Recover reordered norms and their indices for Transient Mode
reordvct.c	Rearrange a vector in decreasing order
weight.c	Calculate quantization weights
cnst.h	Definition of constants
complxop.h	Function prototypes for complex basic operations
proto.h	Function prototypes for codec

<b>Filename</b>	<b>Description</b>
rom.h	Definition of tables
state.h	Definition of state structures of codec

パケットフォーマット、能力識別子及び能力パラメータ

C. 1 参考文献

- [1] IETF RFC 3550 (2003), RTP: A Transport Protocol for Real-Time Applications.
- [2] IETF RFC 5404 (2009), RTP Payload format for G.719

C. 2 標準 J T - G 7 1 9 フレームのパケット構造

本標準で定義されるオーディオ符号化アルゴリズムは、20ms フレーム及び 48kHz のサンプリングレートクロックを使用して、20Hz から 20kHz の帯域幅のフルバンドのオーディオ信号を符号化する。ビットレートは、全ての 20ms フレーム境界で 32kbit/s から 128kbit/s まで変更することが可能である。32kbit/s から 88kbit/s のビットレートは、4 kbit/s の倍数となり、88kbit/s から 128kbit/s のビットレートは、8 kbit/s の倍数となる。符号化アルゴリズムは、全ての 20ms フレーム境界でビットレートを変えることができる。

標準 J T - G 7 1 9 に対する R T P [1] ペイロードフォーマットは、R F C 5 4 0 4 [2] で指定される。このペイロードフォーマットは、コーデックのマルチレート能力をサポートし、フレーム毎にビットレートを変更することが可能である。

R T P ペイロードフォーマットのインタリーブモードを定義する。このモードは、会話型のサービスには推奨されておらず、標準 J T - H 2 4 5 システムでもサポートされない。

C. 3 標準 J T - H 2 4 5 で使用する能力識別子及びパラメータ

GenericCapability は、I T U - T 勧告、及び標準 J T - H 2 4 5 において、ベースラインの標準 J T - G 7 1 9 の能力交換に使用される。ここでは、機能識別子とパラメータについては、必要なテーブルを定義する。

C. 3. 1 標準 J T - G 7 1 9 の能力識別子及びパラメータ

標準 J T - H 2 4 5 能力と OpenLogicalChannel メッセージにおいて、最大ビットレートをシグナリングするために、Table C.2/JT-G719 に示す maxBitRate 領域が使用されなければならない。

Table C.1/JT-G719 Capability identifier table for G.719  
(ITU-T G.719)

Capability name	ITU-T Rec. G.719
Capability class	Audio
Capability identifier type	Standard
Capability identifier value	{ itu-t(0) recommendation(0) g(7) 719 generic-capabilities(1) 0 }
maxBitRate	This parameter shall be set to a value of at least 320, representing 32 kbit/s
Collapsing	This field shall contain the G.719 Capability Parameters as given below.
nonCollapsing	This field shall not be included.
NonCollapsingRaw	This field is not used.
Transport	This field is not used.

例えば、maxBitRate が 480 であり、{ itu-t(0) recommendation(0) g(7) 719 generic-capabilities(1) 0 } の能力識別子が含まれるシステムは、48 kbit/s までのこの標準に応じた動作が可能であることを示す。この能力識別子の maxBitRate 領域は、標準 J T - H 2 4 5 に応じて使用される。maxBitRate 領域の値は、R T P ペイロード仕様

[2]に指定される全てのオーディオ符号化の冗長フレームを含まなければならない。

以下の Table C.2/JT-G719 は、必須な maxFramesPerPacket パラメータを定義する。受信能力においては、これは、受信側で復号が可能な単一の RTP パケット内にある符号化された標準 J T - G 7 1 9 フレームの最大数を示す。

Table C.2/JT-G719 Generic Capability parameter table for G.719, describing  
the maximum number of frames allowed in an RTP packet  
(ITU-T G.719)

Parameter name	maxFramesPerPacket
Parameter description	This is a Collapsing GenericParameter. The value of maxFramesPerPacket specifies the maximum number of encoded G.719 frames that may be included in a single RTP packet.
Parameter identifier value	1
Parameter status	Mandatory
Parameter type	unsignedMin
Supersedes	This field is not used.

以下の Table C.3/JT-G719 にオプションの C B R (固定ビットレート) パラメータを定義する。これは、受信能力において、オーディオ符号化レートが、C B R ブール値の配列で指定された固定のレートで動作することを示す。このパラメータは、標準 J T - H 3 2 0 システムを相互に作用させるために有用である。また、このパラメータにより指定されたレートは、R T P ペイロード仕様[2]で指定された全ての非冗長なオーディオ符号化フレームを含まなければならない。

openLogicalChannel コマンドにおいて、単一のレートのみが指定される。なお、openLogicalChannel コマンドにおいて、NumChannels が 1 チャンネルより大きい場合には 32 と 48 kbit/s は指定してはならない。

Table C.3/JT-G719 Generic Capability parameter table for G.719, describing  
support for constant bit rate  
(ITU-T G.719)

Parameter name	CBR
Parameter description	This parameter is a bit array defined in Table C.4
Parameter identifier value	2
Parameter status	Optional
Parameter type	booleanArray
Supersedes	This field is not used.

Table C.4/JT-G719 CBR parameter Definition  
(ITU-T G.719)

Parameter name	CBR
Parameter description	Bit 8 (value 1) – If set, indicates 32 kbit/s Bit 7 (value 2) – If set, indicates 48 kbit/s Bit 6 (value 4) – If set, indicates 64 kbit/s Bit 5 (value 8) – If set, indicates 96 kbit/s Bit 4 (value 16) – If set, indicates 128 kbit/s  Bits 1-3 are reserved, and shall be set to 0 by transmitters and ignored by receivers.

以下の Table C.5/JT-G719 に、オプション NumChannels パラメータを定義する。受信能力において、このパ



ラメータは受信側でサポートされる最大チャンネル数を示す。このパラメータが省略される場合、最大チャンネル数は1チャンネルを意味する。CBRパラメータが設定される場合、このパラメータの値は2以下でなければならない。また openLogicalChannel コマンドでは、ストリームにおいて使用されるチャンネル数を指定する。

Table C.5/JT-G719 Generic Capability parameter table for G.719, describing  
number of channels supported

(ITU-T G.719)

Parameter name	NumChannels
Parameter description	The parameter indicates the number of channels (e.g. 2 for stereo)
Parameter identifier value	3
Parameter status	Optional
Parameter type	unsignedMin
Supersedes	This field is not used.

付録  
 (標準 J T - G 7 1 9 に対する)  
 用語対照表

英 語	T T C 標準用語
bandwidth extension	帯域拡張
bit allocation	ビット割り当て
bit stream	ビット列
capability identifier	能力識別子
capability parameters	能力パラメータ
folding	折り返し
fullband	フルバンド
Huffman code	ハフマン符号
lattice vector quantization	格子ベクトル量子化
leader	リーダー
modified discrete cosine transform	修正離散コサイン変換
noise-fill	雑音充填
overlap and add	重ね合わせ加算
packet format	パケットフォーマット
payload format	ペイロードフォーマット
spectrum filling	スペクトル充填
strage format	蓄積形式
time domain aliasing	時間域エリアシング