

(19) 日本国特許庁 (JP)

(12) 特許公報 (B2)

(11) 特許番号

**特許第4049793号**

(P4049793)

(45) 発行日 平成20年2月20日 (2008.2.20)

(24) 登録日 平成19年12月7日 (2007.12.7)

(51) Int. Cl.<sup>8</sup>

F I

H 0 3 M 7/36 (2006.01)

H 0 3 M 7/36

H 0 4 N 7/26 (2006.01)

H 0 4 N 7/13

Z

請求項の数 20

(全 37 頁)

(21) 出願番号 特願2005-513652(P2005-513652)  
 (86) (22) 出願日 平成16年9月1日 (2004.9.1)  
 (86) 国際出願番号 PCT/JP2004/012646  
 (87) 国際公開番号 WO2005/025072  
 (87) 国際公開日 平成17年3月17日 (2005.3.17)  
 審査請求日 平成17年8月24日 (2005.8.24)  
 (31) 優先権主張番号 特願2003-310106(P2003-310106)  
 (32) 優先日 平成15年9月2日 (2003.9.2)  
 (33) 優先権主張国 日本国 ( J P )

(73) 特許権者 000004226  
 日本電信電話株式会社  
 東京都千代田区大手町二丁目3番1号  
 (74) 代理人 100121706  
 弁理士 中尾 直樹  
 (74) 代理人 100066153  
 弁理士 草野 卓  
 (74) 代理人 100128705  
 弁理士 中村 幸雄  
 (72) 発明者 守谷 健弘  
 東京都千代田区大手町二丁目3番1号 日  
 本電信電話株式会社内  
 (72) 発明者 ヤン ダイ  
 東京都千代田区大手町二丁目3番1号 日  
 本電信電話株式会社内

最終頁に続く

(54) 【発明の名称】 浮動小数点信号可逆符号化方法、復号化方法、及びそれらの装置、プログラム及びその記録媒体

(57) 【特許請求の範囲】

【請求項 1】

(a) 浮動小数点形式の第 1 信号サンプル系列を整数形式の第 2 信号サンプル系列に変換するステップと、

(b) 上記整数形式の第 2 信号サンプル系列を可逆圧縮して第 1 符号列を生成するステップと、

(c) 上記整数形式の第 2 信号サンプル系列と上記浮動小数点形式の第 1 信号サンプル系列との差分と対応する浮動小数点形式の差分信号を生成するステップと、

(d) 上記浮動小数点形式の差分信号のサンプルにおける、上記整数形式の第 2 信号サンプル系列の対応する各サンプルの絶対値により決まる非ゼロになり得る桁のビット列を符号化する第 1 符号化処理と、上記差分信号の全桁を符号化する第 2 符号化処理とでいずれが高い圧縮効率であるかを判定するステップと、

(e) 上記判定が上記第 1 符号化処理であれば、上記差分信号を上記第 1 符号化処理により符号化して第 2 符号列を生成し、上記判定が上記第 2 符号化処理であれば、上記差分信号を上記第 2 符号化処理により符号化して第 2 符号列を生成するステップ、  
 とを含む浮動小数点信号可逆符号化方法。

【請求項 2】

(a) 浮動小数点形式の第 1 信号サンプル系列をフレーム毎に直接可逆圧縮する場合と、整数値サンプル系列と差分信号の 2 系統信号に分離して圧縮する場合とのいずれが高い圧縮効率となるかを判定するステップと、

(b) 上記判定が2系統信号分離であれば、

(b-1) 上記第1信号サンプル系列を整数形式の第2信号サンプル系列に変換するステップと、

(b-2) 上記整数形式の第2信号サンプル系列を可逆圧縮して第1符号列を生成するステップと、

(b-3) 上記整数形式の第2信号サンプル系列と上記浮動小数点形式の第1信号サンプル系列との差分と対応する浮動小数点形式の差分信号を生成するステップと、

(b-4) 上記浮動小数点形式の差分信号から第2符号列を生成するステップ、  
とにより2系統信号分離圧縮を行うステップと、

(c) 上記判定が直接可逆圧縮であれば、上記第1信号サンプル系列を直接可逆圧縮して第3符号列を生成するステップと、 10

(d) 上記直接可逆圧縮したか上記2系統信号分離圧縮したかを示す補助符号を生成するステップ、

とを含む浮動小数点信号可逆符号化方法。

【請求項3】

請求項2記載の浮動小数点信号可逆符号化方法において、上記ステップ(b-1)の上記整数形式の第2信号サンプル系列への変換は小数点以下の切り捨てにより行われ、

上記ステップ(b-4)は、

(b-4-1) 上記浮動小数点形式の差分信号のサンプルにおける、上記整数形式の第2信号サンプル系列の対応する各サンプルの絶対値により決まる非ゼロになり得る桁のビット列を符号化する第1符号化処理と、上記差分信号の全桁を符号化する第2符号化処理とでいずれが高い圧縮効率であるかを判定するステップと、 20

(b-4-2) 上記判定が上記第1符号化処理であれば、上記差分信号を上記第1符号化処理により符号化して第2符号列を生成し、上記判定が上記第2符号化処理であれば、上記差分信号を上記第2符号化処理により符号化して第2符号列を生成するステップ、  
とを含む。

【請求項4】

請求項1又は3記載の浮動小数点信号可逆符号化方法において、上記第1符号化処理は、上記非ゼロとなり得る桁のビット列を可逆圧縮して上記第2符号列を生成する場合と、上記非ゼロになり得るビット列をそのまま上記第2符号列として出力する場合のいずれか一方を実行する。 30

【請求項5】

請求項1又は3記載の浮動小数点信号可逆符号化方法において、上記第2符号化処理は、上記差分信号の仮数部を可逆圧縮して上記第2符号列を生成する場合と、上記仮数部をそのまま上記第2符号列として出力する場合のいずれか一方である。

【請求項6】

請求項1又は3記載の浮動小数点信号可逆符号化方法において、上記整数形式の第2信号サンプル系列の1サンプルのビット長 $b_1$ は上記浮動小数点形式の第1信号サンプル系列の仮数部のビット数 $b_M$ より小であり、上記第1符号化処理は、上記非ゼロになり得る桁のビット列を、上記仮数部のビット数 $b_M$ と上記1サンプルのビット長 $b_1$ の差のビット数の上位側と、残りの下位側とに分けてそれぞれ可逆圧縮してそれぞれの符号列を得て、上記得られた符号列を合わせて上記第2符号列として出力する。 40

【請求項7】

請求項1又は3記載の浮動小数点信号可逆符号化方法において、

上記第1符号化処理は、複数サンプルのフレーム毎に行われ、フレーム内の上記差分信号のサンプルの仮数部における非ゼロとなり得るビット領域である端数部分を最上位ビット又は最下位ビットをそろえて配置し、少なくとも時間方向および/又は振幅方向に複数のビットを選択して1つ又は複数の走査ビット列を生成し、上記走査ビット列を可逆圧縮して上記第2符号列を生成するステップを含む。

【請求項8】

請求項1又は3記載の浮動小数点信号可逆符号化方法において、上記第1符号化処理は、複数サンプルのフレームごとに行われ、フレーム内の上記差分信号の各サンプルにおける、非ゼロとなり得る桁のビット列を配列し、上記ビット列の配列を予め決めた複数の異なる走査順序で走査して、それぞれの走査ビット列を生成するステップと、各上記走査順序による上記走査ビット列を符号化してそれぞれの走査順序による符号列を生成するステップと、どの走査順序による符号列が最も情報量が少ないかを判定し、判定した符号列を上記第2符号列として出力すると共に、対応する走査順序を表す補助符号を出力するステップとを含む。

【請求項9】

請求項1又は3記載の浮動小数点信号可逆符号化方法において、上記第1符号化処理は、複数サンプルのフレーム毎にフレーム内の上記差分信号の端数部分を整数値サンプルとするサンプル列の平均値を求め、それぞれの整数値サンプルの値から上記平均値を減算した信号に対し、時間方向に予測を行い、その予測誤差を可逆圧縮符号化するとともに、上記平均値を符号化して上記第2符号列とするステップを含む。

【請求項10】

(a) 入力補助符号に応じて入力符号列が1系統符号化であるか2系統符号化であるかを判定するステップと、

(b) 上記判定において1系統符号化であると判定された場合は、入力符号列を1つの符号列として復号伸張して浮動小数点形式の原信号サンプル系列を生成するステップと、

(c) 上記判定において2系統符号化であると判定された場合は、入力符号列を第1符号列及び第2符号列とに分離するステップと、

(d) 上記第1符号列を復号伸張して整数形式の第1信号サンプル系列を生成するステップと、

(e) 上記第2符号列から浮動小数点形式の差分信号を得るステップと、

(f) 上記整数形式の第1信号サンプル系列を浮動小数点形式の第2信号サンプル系列に変換するステップと、

(g) 上記浮動小数点形式の第2信号サンプル系列と上記浮動小数点形式の差分信号を合成して浮動小数点形式の原信号サンプル系列を生成するステップ、

とを含む浮動小数点信号復号化方法。

【請求項11】

請求項10記載の浮動小数点信号復号化方法において、上記ステップ(e)は、上記整数形式の第1信号サンプル系列又は入力補助符号に応じて上記第2符号列の復号伸張を第1伸張法によるか第2伸張法によるかを判定するステップと、

この判定が第1伸張法であれば、上記第1信号サンプル系列の各サンプルの絶対値に応じて一意に決まる非ゼロになり得る桁数のビット列に、上記第2符号列を復号伸張し、その復号伸張されたビット列から上記差分信号を組み立て生成し、上記判定が第2伸張法であれば、その伸張法に応じて上記第2符号列を第2伸張法により上記差分信号に直接復号伸張するステップ、

とを含む。

【請求項12】

(a) 第1符号列を復号伸張して整数形式の第1信号サンプル系列を生成するステップと、

(b) 上記第1信号サンプル系列又は補助符号に応じて第2符号列の復号伸張を第1伸張法によるか、第2伸張法によるかを判定するステップと、

(c) 上記判定が第1伸張法であれば上記整数形式の第1信号サンプル系列の各サンプルの絶対値に応じて一意に決まる非ゼロになり得る桁数のビット列に、上記第2符号列を復号伸張し、その復号伸張されたビット列を浮動小数点形式の差分信号に組み立てるステップと、

(d) 上記判定が第2伸張法であれば上記第2符号列から上記浮動小数点形式の差分デジタル信号の全桁を生成するステップと、

(e) 上記整数形式の第1信号サンプル系列を浮動小数点形式の第2信号サンプル系列に変換するステップと、

(f) 上記浮動小数点形式の第1信号サンプル系列と上記浮動小数点形式の差分信号を合成して浮動小数点形式の原デジタル信号系列を生成するステップ、  
とを含む浮動小数点信号復号化方法。

【請求項13】

請求項11又は12記載の浮動小数点信号復号化方法において、上記第1伸張法による上記差分信号の生成ステップは、

上記整数形式の第1信号サンプル系列の各サンプルの絶対値に応じて一意に決まる非ゼロになり得る桁数のビット列に上記第2符号列中の第1部分を復号伸張するステップと、 10

上記整数形式の第1信号サンプル系列のサンプルのビット数と上記浮動小数点形式の原信号サンプル系列の仮数部のビット数との差に応じて一意に決まる非ゼロになり得る桁数に上記第2符号列中の第2部分を復号伸張するステップと、

これら第2符号列中の第1部分を復号伸張したものと第2符号列中の第2部分を復号伸張したものとを合成して上記差分信号を生成するステップ、  
とを含む。

【請求項14】

入力された浮動小数点形式の第1信号サンプル系列中の各サンプルを、整数値とその誤差に分離して可逆圧縮符号化する第1符号化手段と、

上記浮動小数点形式の第1信号サンプル系列を直接可逆圧縮符号化する第2符号化手段と、 20

上記浮動小数点形式の第1信号サンプル系列のフレーム毎に、上記第1符号化手段と上記第2符号化手段とのいずれの圧縮効率が高いかを判定し又は圧縮効率が高いかを推定し、その高いと判定又は推定した方の符号化手段を選択し、その選択情報を示す補助符号を生成する分析選択手段、

とを含む浮動小数点信号可逆符号化装置。

【請求項15】

入力された浮動小数点形式の第1信号サンプル系列を整数形式の第2信号サンプル系列に変換して可逆圧縮する第1符号化部と、

上記整数形式の第2信号サンプル系列と上記浮動小数点形式の第1信号サンプル系列との差分と対応する浮動小数点形式の差分信号を生成する差分生成部と、 30

上記浮動小数点形式の差分信号の全桁を符号化する第2符号化部と、

上記差分信号のサンプルにおける、上記整数形式の信号サンプル系列の対応する各サンプルの絶対値で決まる非ゼロになり得る桁のビット列を符号化する第3符号化部と、

上記第2符号化部と上記第3符号化部とのいずれの圧縮効率が高いかを判定し又は圧縮効率が高いかを推定し、その高いと判定又は推定した方の符号化部による符号を選択出力する手段、

とを含む浮動小数点信号可逆符号化装置。

【請求項16】

入力された符号列を、浮動小数点形式の信号サンプル系列に直接可逆復号伸張する第1復号手段と、 40

入力された符号列を整数値とその誤差に分離して可逆復号伸張して、浮動小数点形式デジタル信号系列を生成する第2復号手段と、

入力された補助符号に応じて、上記第1復号手段と上記第2復号手段の一方を選択する選択手段、

とを含む浮動小数点信号復号化装置。

【請求項17】

入力された第1符号列を復号伸張して整数形式の第1信号サンプル系列を生成する第1伸張部と、

上記差分信号のサンプルにおける、上記整数形式の第1信号サンプル系列の対応する各 50

サンプルの絶対値で決まる非ゼロになり得る桁数に入力された第 2 符号列を復号伸張し、その復号伸張されたビット列を浮動小数点形式の差分信号に組み立てる第 2 伸張部と、

上記入力された第 2 符号列を上記浮動小数点形式の差分信号に直接復号伸張する第 3 伸張部と、

上記第 1 信号サンプル系列又は入力された補助符号に応じて上記第 2 伸張部と上記第 3 伸張部との一方を選択する選択部と、

上記浮動小数点形式の第 1 信号サンプル系列と、上記第 2 伸張部及び第 3 伸張部中の上記選択されたものからの上記浮動小数点形式の差分信号とを合成して浮動小数点形式の第 3 信号サンプル系列を生成する合成部、

とを含む浮動小数点信号復号装置。

10

#### 【請求項 18】

請求項 1 乃至 9 のいずれかに記載した浮動小数点信号可逆符号化方法の各過程をコンピュータに実行させるための符号化プログラム。

#### 【請求項 19】

請求項 10 乃至 13 のいずれかに記載した浮動小数点形式信号系列復号化方法の各過程をコンピュータに実行させるための復号化プログラム。

#### 【請求項 20】

請求項 18 又は 19 に記載したプログラムを記録したコンピュータ読み取り可能な記録媒体。

#### 【発明の詳細な説明】

20

#### 【技術分野】

#### 【0001】

この発明は音声、音楽、画像などの浮動小数点形式の信号サンプル系列を、歪を許さずより低い情報量に圧縮された符号に変換する符号化方法、その復号化方法、符号化装置、復号化装置、これらのプログラム、記録媒体に関する。

#### 【背景技術】

#### 【0002】

音声、画像などの情報を圧縮する方法として、歪を許さない可逆な符号化方法がある。

圧縮率の高い非可逆の符号化を行い、その再生信号と原信号の誤差を可逆に圧縮することを組み合わせることで高い圧縮率で可逆な圧縮が可能となる。この組み合わせ圧縮方法が特許文献 1 に提案されている。この方法は前記文献に詳細に示されているが、以下に簡単に説明する。

30

符号器では、デジタル入力信号（以下、入力信号サンプル系列とも呼ぶ）が、フレーム分割部で、その入力信号サンプル系列が、例えば 1024 個の入力信号サンプルからなるフレーム単位に、順次分割され、このフレーム単位毎にデジタル信号が非可逆圧縮符号化される。この符号化は、復号化時に元のデジタル入力信号をある程度再現できる方式であれば、入力信号に適した如何なる方式でもよい。

#### 【0003】

例えば、上記デジタル入力信号が音声であればITU-のG.729標準として勧告されている音声符号化などが利用でき、音楽であればMPEG-4で採用されているTwinVQ(Transform-Domain Weighted Interleaved Vector Quantization)符号化などが利用できる。この非可逆圧縮符号は局部復号され、この局部信号と元のデジタル信号との誤差信号が生成される。なお、実際的には、局部復号はする必要なく、非可逆圧縮符号を生成する際に得られる量子化信号と元のデジタル信号との誤差を求めればよい。この誤差信号の振幅は通常は元のデジタル信号の振幅よりもかなり小さい。よって、元のデジタル信号を可逆圧縮符号化するよりも、誤差信号を可逆圧縮符号化の方が情報量を小さくできる。

40

#### 【0004】

この可逆圧縮符号化の効率を上げるために、この誤差信号の符号絶対値表現（極性と絶対値の2進数）されたサンプル列のフレーム内の全サンプルに対しそれらの各ビット位置

50

、つまりMSB, 第2MSB, ..., LSB毎に、サンプル系列方向(時間方向)にビットを連結したビット列を生成する。即ち、ビット配列の変換が行われる。この各連結した同じビット位置の1024個のビットからなるビット列をここでは便宜上「等位ビット列」と呼ぶことにする。これに対し、各サンプルの極性も含む振幅値を表す1ワードのビット列を便宜上「振幅ビット列」と呼ぶことにする。誤差信号は振幅が小さいので、各サンプルの最上位から1つ又は連続する複数のビットは全て“0”となることが多い。そこで、それらのビット位置で連結して生成した等位ビット列を予め決めた符号で表すことによつて、誤差信号の可逆圧縮符号化効率を上げることができる。

#### 【0005】

これら等位ビット列が可逆圧縮符号化される。可逆圧縮符号化としては、例えば、同一符号(1又は0)が連続する系列がある場合や頻繁に出現する系列がある場合を利用した、ハフマン符号化や算術符号化などのエントロピー符号化などを用いることができる。

復号化側では、可逆圧縮符号が復号化され、その復号信号に対し、ビット配列の逆変換が行われ、即ち、フレーム毎に等位ビット列を振幅ビット列に変換し、得られた誤差信号が順次再生される。また、非可逆圧縮符号が復号化され、この復号信号と再生された誤差信号とが加算され、最後に、フレームごとの各加算信号が順次連結されて、元のデジタル信号系列が再生される。

#### 【0006】

音声、画像などの情報を歪を許さない可逆な符号化方法としてはその他にも各種のものが知られている。音楽情報については例えば非特許文献1に示されている。従来の方法は何れも波形をそのままPCM信号としたものについての圧縮符号化方法であった。

しかし音楽の収録スタジオでは浮動小数点形式で波形が記録されて保存されることがある。浮動小数点形式の値は極性、指数部、仮数部に分離されている。例えば図37に示すIEEE-754として標準化されている浮動小数点形式は32ビットであり、上位ビットから極性1ビット、指数部8ビット、仮数部23ビットで構成されている。極性をS、指数部の8ビットで表す値を10進数でE、仮数部の2進数をMとすると、この浮動小数点形式の数値は絶対値表現2進数で表わすと

#### 【0007】

[数1]

$$(-1)^S \times 1.M \times 2^{E-E_0} \quad (1)$$

#### 【0008】

IEEE-754によれば、 $E = 2^7 - 1 = 127$ と決められており、これにより、式(1)中の $E - E_0$ は

$$-127 \leq E - E_0 \leq 128$$

の範囲の任意の値を取ることができる。ただし、 $E - E_0 = -127$ の場合は $a110$ 、 $E - E_0 = 128$ の場合は $a111$ と定義されている。 $E - E_0 = n$ は式(1)で表される値の整数部分の桁数(ビット数)から1を減算した値、即ち、最上位の“1”より下位のビット数を表わしている。

【特許文献1】特開2001-44847号公報

【非特許文献1】Mat Hans及びRonald W. Schaffer著「Lossless Compression of Digital Audio」IEEE SIGNAL PROCESSING MAGAZINE, JULY 2001, P21~32 音声、音楽、画像の情報が浮動小数点形式のデジタル信号系列とされている場合は、浮動小数点形式の性質上“0”と“1”からなるビット列が乱雑な場合が多くなるため、前述したビット配列変換を行っても、エントロピー圧縮符号化などによって高い圧縮率は期待できない。また浮動小数点形式のサンプル系列は原アナログ波形と著しく異なったものとなりサンプル間の相関による冗長性がなく、非特許文献1に示す可逆予測符号化

方法を適用しても高い圧縮率は期待できない。

【発明の開示】

【発明が解決しようとする課題】

【0009】

この発明の目的は浮動小数点形式の信号サンプル系列に対して圧縮率が高く、歪が生じない可逆符号化、復号化方法、それらの装置及びそれらのプログラム、記録媒体を提供することにある。

【課題を解決するための手段】

【0010】

この発明の第1の観点による浮動小数点信号可逆符号化方法は、  
 (a) 浮動小数点形式の第1信号サンプル系列を整数形式の第2信号サンプル系列に変換するステップと、

(b) 上記整数形式の第2信号サンプル系列を可逆圧縮して第1符号列を生成するステップと、

(c) 上記整数形式の第2信号サンプル系列と上記浮動小数点形式の第1信号サンプル系列との差分と対応する浮動小数点形式の差分信号を生成するステップと、

(d) 上記浮動小数点形式の差分信号のサンプルにおける、上記整数形式の第2信号サンプル系列の対応する各サンプルの最上位の1のビット位置により決まる非ゼロになり得る桁のビット列を符号化する第1符号化処理と、上記差分信号を直接符号化する第2符号化処理とでいずれが高い圧縮効率であるかを判定するステップと、

(e) 上記判定が上記第1符号化処理であれば、上記差分信号を上記第1符号化処理により符号化して第2符号列を生成し、上記判定が上記第2符号化処理であれば、上記差分信号を上記第2符号化処理により符号化して第2符号系列を生成するステップ、  
 とを含む。

【0011】

この発明の第2の観点による浮動小数点信号可逆符号化方法は、

(a) 浮動小数点形式の第1信号サンプル系列をフレーム毎に直接圧縮する場合と、整数値サンプル系列と差分信号の2系統信号に分離して圧縮する場合とのいずれが高い圧縮効率となるかを判定するステップと、

(b) 上記判定が2系統信号分離であれば、

(b-1) 上記第1信号サンプル系列を整数形式の第2信号サンプル系列に変換するステップと、

(b-2) 上記整数形式の第2信号サンプル系列を可逆圧縮して第1符号列を生成するステップと、

(b-3) 上記整数形式の第2信号サンプル系列と上記浮動小数点形式の第1信号サンプル系列との差分と対応する浮動小数点形式の差分信号を生成するステップと、

(b-4) 上記浮動小数点形式の差分信号から第2符号列を生成するステップ、

とにより2系統信号分離圧縮を行うステップと、

(c) 上記判定が直接圧縮であれば、上記第1信号サンプル系列を直接可逆圧縮して第3符号列を生成するステップと、

(d) 上記直接可逆圧縮したか上記2系統信号分離圧縮したかを示す補助符号を生成するステップ、

とを含む。

【発明の効果】

【0012】

この発明の第1の観点によれば、浮動小数点形式の入力信号サンプル系列Xの平均振幅が所定値より大きい場合には、信号サンプル系列Xを直接可逆符号化し、所定値以下の場合には入力信号サンプル系列Xを整数サンプル系列と差分信号に分けて符号化することにより、効率的な符号化が可能である。

この発明の第2の観点によれば、浮動小数点形式の入力信号サンプル系列Xを整数サン

ブル系列 Y と差分信号 Z に分け、それぞれ符号化する場合に、差分信号については非ゼロとなり得る桁のみを符号化する場合と、差分信号をそのまま符号化する場合で圧縮効率の高いほうの符号化結果を用いることにより、さらに効率的な符号化が可能である。

【図面の簡単な説明】

【0013】

- [図1] この発明の第1実施形態の符号化装置の機能構成を示す図。
- [図2] 図2中の整数化部12の動作を説明するための図。
- [図3] 図2中の整数化部12における処理手順の例を示す図。
- [図4] 整数値の絶対値と、その値を示す浮動小数点の仮数部のビットとの対応を示す図。
- [図5] 整数値の絶対値と、差分デジタル信号の極性、指数部および仮数部との関係を示す図。
- [図6] 差分信号中の圧縮符号化される部分の例を示す図。
- [図7] この発明の第1実施形態の復号化装置の機能構成を示す図。
- [図8] 図1に示す符号化装置の変形実施形態を示す図。
- [図9] この発明の第2実施形態の符号化装置の機能構成を示す図。
- [図10] この発明の第2実施形態の復号化装置の機能構成を示す図。
- [図11] この発明の第2実施形態の符号化装置の変形機能構成を示す図。
- [図12] この発明の第3実施形態の符号化装置の機能構成を示す図。
- [図13] この発明の第3実施形態の復号化装置の機能構成を示す図。
- [図14A] もともと浮動小数点信号であった入力浮動小数点形式信号 X に対する16ビット整数化を説明するための図。
- [図14B] 24ビット整数値信号から生成された浮動小数点形式信号 X に対する16ビット整数化を説明するための図。
- [図14C] 16ビット整数値信号から生成された浮動小数点信号 X に対する16ビット整数化を説明するための図。
- [図15A] 図9Aの場合の仮数部の非ゼロとなり得る領域を示す図。
- [図15B] 図9Bの場合の仮数部の非ゼロとなり得る領域を示す図。
- [図16] この発明の第4実施形態の符号化装置の機能構成を示す図。
- [図17] この発明の第4実施形態の復号化装置の機能構成を示す図。
- [図18] 仮数部を時間方向にビット走査して符号化する動作を説明するための図。
- [図19] 図18の符号化を実行する不圧縮部17の構成例を示す図。
- [図20] 図19の符号化に対応した復号化装置における伸張部23の構成例を示す図。
- [図21] 時間方向のビット走査の他の例を示す図。
- [図22] 時間方向のビット走査の他の例を示す図。
- [図23] 時間方向のビット走査の更に他の例を示す図。
- [図24] 非ゼロとなり得るビット列をLSB側に位置をそろえて配列した場合の時間方向のビット走査を示す図。
- [図25] 第6実施形態の符号化装置における圧縮部17の構成例を示す図。
- [図26] 第6実施形態の復号化装置における伸張部23の構成例を示す図。
- [図27] 第7実施形態の符号化装置における圧縮部17の構成例を示す図。
- [図28] 第7実施形態の復号化装置における伸張部23の構成例を示す図。
- [図29] 図27の圧縮部17の変形例を示す図。
- [図30] 図11の符号化装置における圧縮部17と131の代わりに使用可能な構成を示す図。
- [図31] 図30の構成に対応して図10の復号化装置における伸張部23、231、選択部233、補助復号部234に置き換える構成を示す図。
- [図32] 浮動小数点入力と切捨てによる整数値との関係を示す図。
- [図33] 浮動小数点入力と四捨五入による整数値との関係を示す図。
- [図34] 浮動小数点入力と変形四捨五入による整数値との関係を示す図。



[ 図 3 5 ] 図 1 1 の符号化装置の変形例を示す図。

[ 図 3 6 ] この発明をコンピュータで実施する場合の構成を示す図。

[ 図 3 7 ] IEEE - 7 5 4 の 3 2 ビット浮動小数点のフォーマットを示す図。

【発明を実施するための最良の形態】

【 0 0 1 4 】

この発明は画像信号、音声信号などにも適用できるが、以下は主として音楽信号に対してこの発明を適用した場合について、実施形態を説明する。また浮動小数点形式としても各種のものを用いることができるが、以下では 3 2 ビット IEEE - 7 5 4 を前提とする。

### 第 1 実施形態

図 1 にこの発明による符号化装置の実施形態の機能構成を示す。この実施形態の符号化装置は、分析選択部 1 0 0 と、1 系統符号化部 1 2 0 と、2 系統符号化部 1 1 0 と、補助符号化部 1 0 3 と、出力部 1 0 4 とから構成されている。分析選択部 1 0 0 は効率推定部 1 0 1 と選択部 1 0 2 とから構成されている。2 系統符号化部 1 1 0 は、整数化部 1 2 と、圧縮部 1 3、1 7 と、差分生成部 1 4 とから構成されている。

【 0 0 1 5 】

信号源 1 0 から例えば音楽信号のサンプル列が 3 2 ビットの浮動小数点形式の入力デジタル信号サンプル系列 X として出力される。この入力信号サンプル系列 X は 2 4 ビットの整数形式で録音された信号の素材を変形、振幅調整、効果付加、混合などの加工をした結果、小数点以下の数を含むようになったものを浮動小数点形式に変換したもの、あるいは、もともと 2 4 ビットの整数形式で録音された信号を 3 2 ビット浮動小数点形式に変換後、前記加工をした信号である。この入力信号サンプル系列 X は一定サンプル数、例えば 1 0 2 4 個単位のフレームにフレーム分割部 1 1 で分割される。このフレーム分割は 1 6 ~ 3 2 フレームなどのスーパーフレーム単位でもよく、あるいは 1 / 2 フレームとか 1 / 4 フレームでもよく、要は一定サンプル数毎に分割することであり、以下ではこれらを総称してフレームと呼ぶ。

【 0 0 1 6 】

入力信号サンプル系列 X は分析選択部 1 0 0 により分析され、整数値とその誤差を分離して可逆圧縮符号化する 2 系統符号化部 1 1 0 又は入力信号サンプル系列 X を直接可逆圧縮符号化する 1 系統符号化部 1 2 0 のいずれかに切り替えられて与えられる。分析選択部 1 0 0 では例えば入力信号サンプル系列 X が効率推定部 1 0 1 に入力され、そのフレーム内の平均振幅が平均部 1 0 1 a で計算され、その平均振幅が所定値以上か否かが判定部 1 0 1 b で判定され、その判定結果に応じて選択部 1 0 2 が制御される。その制御により判定結果が所定値以上であればそのフレームの入力信号サンプル系列 X が 2 系統符号化部 1 1 0 に供給され、所定値以上でなければ 1 系統符号化部 1 2 0 へ供給される。

【 0 0 1 7 】

例えば入力信号サンプル系列 X の各サンプル中の整数部が 1 6 ビットの 2 進数で表わされる場合、前記判定部 1 0 1 b での判定基準となる所定値は 8 ビットの値とされる。この所定値は前記整数値により異なり、また入力信号サンプル系列の性質などにより、全体の圧縮効率がよくなるように選定される。

2 系統符号化部 1 1 0 においては、浮動小数点形式の信号サンプル系列 X は整数化部 1 2 に入力され、各サンプル毎に整数形式の信号サンプル系列 Y に変換される。先に述べたように前記例では、もともと 2 4 ビットの整数形式の信号の素材を変形、振幅調整、効果付加などの加工をした場合が多く、振幅が大きく変化することがないのが一般的である。従って整数化部 1 2 においては小数点以下を四捨五入または切り捨て、切り上げによって整数に変換すればよい。

【 0 0 1 8 】

例えば切り捨てを行う場合について、図 2 を参照して説明する。整数形式の桁数は 1 6 ビットあるいは 2 0 ビットなどでもよいが、以降の説明では 2 4 ビットを使用する。

浮動小数点による波形の表現法として整数値の 3 2 7 6 8 ( =  $2^{15}$  ) を 1 . 0 と正規

10

20

30

40

50

化して表わす場合、あるいは整数値  $2^{-3}$  を  $1.0$  と正規化して表す場合などもあるが、指数部の値が異なるだけで以下の説明と本質的な差異はない。

小数点の次の上位側ビットは、指数部の値  $E$  が例えば  $E = 147$  であれば仮数部  $M$  の  $MSB$  側から  $147 - 127 = 20$  番目であり、図 2 A に示すように 23 ビットの仮数部  $M$  中の上位側 20 ビット ( $M_1$ ) が絶対値 2 進表現をした場合の整数部分の最上位の 1 より下位の部分であって、仮数部  $M$  中の下位側 3 ビット ( $M_2$ ) が絶対値 2 進表現をした場合の小数点以下の端数部分である。以下では  $M_1$  の部分を仮数部  $M$  における整数部分と呼ぶ。従って、図 2 B に示すように仮数部における整数部分 ( $M_1$ ) 中の最下位ビットが仮数部 23 ビット中の最下位ビットとなるように、仮数部  $M$  を 3 ビット下位方向にシフトすることによりこの小数点以下 3 ビット ( $M_2$ ) が溢れて切り捨てられる。このシフトにより仮数部  $M$  の空き (この例では  $000$ ) となった最上位 3 ビット中の最下位ビット (即ち、23 ビットの最下位から 21 ビット目) に式 (1) における  $1.M$  の 1 に対応する  $1$  を設定することにより、端数切り捨てによる整数化された値が得られる (図 3 C 参照)。あるいは、シフト前の 23 ビットの  $MSB$  の上位側に  $1$  を付加した 24 ビットを 3 ビット下位にシフトしてもよい。

【0019】

更に、このようにして得られた整数値を 2 の補数表現に変換する。つまり、浮動小数点形式の信号サンプル系列  $X$  の各サンプルの極性ビット  $S$  はそのまま最上位ビットとし、残りの 23 ビットについては図 2 C に得られた 23 ビットを、極性  $S$  が  $0$  (正) の場合は図 2 C の 23 ビットをそのまま用い、 $S$  が  $1$  (負) であれば、図 2 D に示すように 23 ビットを全て論理反転、即ち  $0$  と  $1$  を交換した後に最下位ビットに  $1$  を加算する。最上位ビットは極性  $S$  をそのまま用い、図 2 E に示す 24 ビットの補数表現された整数形式の信号サンプル系列  $Y$  が得られる。

【0020】

上述においては、主として、仮数部  $M$  に小数点以下の端数部分と対応するビットが 0 ビット以上含まれている場合を想定し、1 サンプルの浮動小数点形式の 32 ビットデジタル信号を 1 サンプル 24 ビットの整数形式のデジタル信号に変換した。しかし、例えば前記 24 ビット整数形式の信号の素材の複数を混合などして加工する場合には 1 サンプルの振幅値が 24 ビットで表わされる最大値より可成り大きくなっている場合もある。このような  $E = 150$  の場合は例外処理として、指数部  $E$  の値を前記例では  $E = 127 + 23 = 150$  に固定し、例外信号  $y$  を差分生成部 14 に与えると共に、仮数部  $M$  の隣接上位側に  $1$  を付加した  $1M$  の上位 23 ビットをそのまま 2 の補数表現に変換して 24 ビット整数形式の信号サンプル系列  $Y$  とする。

【0021】

整数化部 12 では例えば図 3 に示す処理を行うことになる。まず指数部  $E$  が 150 以上か、つまり整数部が 23 ビット以上であるかを調べ ( $S1$ )、23 ビットより少なければ、桁数計算部 12 A で仮数部における整数部分の桁数  $E - E_0 = n$  を求めて出力し ( $S2$ )、また仮数部  $M$  を下位に  $(23 - n)$  ビットだけシフトして端数部分を溢れ出し、得られた整数部分  $n$  ビットの最下位から  $(n + 1)$  ビット目に  $1$  を付加し ( $S3$ )、更にその上位側に全体で 23 ビットとなるように  $0$  を補充したものを、極性  $S$  を加味して 2 の補数表現の 24 ビット整数形式に変換して整数形式信号サンプル  $Y$  を得る ( $S4$ )。つまり、最上位ビットに極性  $S$  をそのまま用い、残りの 23 ビットは、極性  $S$  が  $0$  (正) であればステップ  $S3$  でシフトして下位から  $(n + 1)$  ビット目に  $1$  を加えた整数部分の最下位ビットから 23 ビットまでをそのまま用い、 $S$  が  $1$  (負) であれば前記シフトして  $1$  を加えた整数部の最下位ビットから 23 ビットまでを、ビット反転して最下位ビットに  $1$  を加算して補数値を得る。ステップ  $S1$  で指数部  $E$  が 150 以上であれば、ステップ  $S5$  でその指数部  $E$  を 150 にしてステップ  $S4$  で  $1M$  の上位 23 ビットから整数形式の信号サンプル  $Y$  を生成する。

【0022】

なお、上述では仮数部  $M$  に対し  $(23 - n)$  ビットシフトして 1 サンプル 24 ビットの

絶対値表現 2 進数の整数値を求めたが、仮数部 M の図 2 A に  $M_1$  で示す上位から  $n = E - E_0$  ビットを取り出し、その上位側に図 2 C に示すように 1 を付加して  $n + 1$  ビットとし、その上位側に  $(2^2 - n)$  ビットの 0 を付加して 23 ビットとし、更にその上位側に極性ビット S を付加して全体で 24 ビットにしてもよい。

以上のようにして整数化部 12 で変換された整数値信号サンプル系列 Y は圧縮部 13 で整数値として波形値の相関などを利用した効率のよい可逆圧縮法により圧縮符号化されて符号列 C a として出力される。圧縮部 13 における可逆圧縮は例えば前記非特許文献 1 に示されているように各サンプル毎に予測値 (整数値) との差分を求め、その差分の系列を、従来の技術の項で述べたようにビット配列変換を行った後、つまり等位ビット列についてエントロピー符号化すればよい。つまり整数値信号サンプル系列 Y は信号源 11 の浮動小数点形式の入力信号サンプル系列 X の原アナログ信号波形と近似したものとなっている。従ってこの入力信号サンプル系列 X は予測や変換を使って信号サンプル間の相関による冗長性を除くことにより効率の高い可逆圧縮符号化が可能となる。

#### 【0023】

差分生成部 14 では整数値信号サンプル Y と信号源 11 より対応する浮動小数点形式の入力信号サンプル X との差分信号 (誤差) Z が生成される。この例では整数値信号サンプル Y は浮動小数点化部 15 で浮動小数点形式の信号サンプル Y' に再変換され、その再変換された浮動小数点形式の信号サンプル Y' が原浮動小数点形式の入力信号サンプル X から減算部 16 で減算されて浮動小数点形式の差分信号 Z が生成される。

浮動小数点化部 15 での変換は 1 サンプルの整数形式デジタル信号が 24 ビットの場合、まったくあいまい性や例外なく 32 ビットの浮動小数点形式の入力信号に変換可能である。前述したように原浮動小数点形式の信号サンプル系列 X の指数部 E は 149 以下の場合が多く、この場合は信号サンプル系列 Y' のサンプルと原浮動小数点形式の入力信号サンプル系列 X のサンプルとの差分信号 Z のサンプルは、原入力信号サンプル系列 X のサンプルの小数点以下の端数値と等しい。

#### 【0024】

上述の差分生成部 14 の説明は理解のためのものであって、現実には、整数値信号サンプル系列 Y を浮動小数点化する必要はなく、差分生成部 14 に入力サンプル系列 X とその仮数部の整数部分桁数  $n$  を入力し、例外信号  $y$  が与えられないときは、浮動小数点形式の入力サンプル系列 X の各サンプルの極性 S と指数部 E、仮数部 M の下位  $(2^3 - n)$  ビットをそれぞれ差分信号 Z の各サンプルの極性、指数部、仮数部の下位  $(2^3 - n)$  ビットとし、仮数部中の上位  $n$  ビットをすべて 0 とする。なお、仮数部の整数部分の桁数  $n = E - E_0$  は、整数値信号サンプル系列 Y のサンプルの最上位の 1 より下位の桁数と等しい。整数値信号サンプルのビット長は固定されているので、最上位の 1 からそれより上位の桁数が決まっても、仮数部中の整数部分の桁数が決まる。即ち、整数信号サンプルの最上位の 1 のビット位置がわかれば、対応する浮動小数点信号サンプルの仮数部における整数部分の桁数がわかる。例外信号  $y$  が与えられた場合は、極性は入力サンプルの極性 S、指数部は  $E - 150$ 、仮数部は入力サンプル系列 X のサンプルと信号サンプル系列 Y' のサンプルの仮数部の差分とする。

#### 【0025】

圧縮部 17 は、整数値信号サンプル系列 Y と浮動小数点形式の入力サンプル系列 X との、対応サンプルごとの差分信号 Z、つまり差分生成部 14 からの浮動小数点形式の差分信号 Z を可逆圧縮符号化し、符号列 C b を出力する。この圧縮部 17 では、例えば、整数化部 12 から出力された仮数部の整数部分桁数  $n$  を用いて、浮動小数点形式の差分信号 Z 中の各サンプルについて、0 以外になり得る桁についてのみ可逆圧縮符号化を行う。浮動小数点形式の入力サンプル系列 X の各サンプルを、

$$X = 1 \cdot M \times 2^{-n} = X_w \cdot X_f$$

ただし  $X_w = 1 M_E$  ( $M_E$  より隣接上位側に 1 が付加された値) は 2 進整数部分、 $X_f = M_F$  は端数部分、とおくと、その仮数部 M における整数部分  $M_E$  に対応するビットと、端数部分 (小数值)  $M_F$  に対応するビットは図 4 に示す関係にある。ただし、仮数部 M は式

(1)におけるMの上位側隣接1ビットを合わせて24ビットの値として示し、絶対値表現2進数とした場合の小数点P<sub>0</sub>の位置を便宜的に示してある。

【0026】

整数値の絶対値 $X_w$ が0の場合は、前式(1)において $E - E_0 = -127$ の場合と、 $-126$   $E - E_0 = -1$ の場合がある。前者の場合は $E = 0$ の特殊な場合であり、仮数部Mの全23ビット $x_{23} \dots x_1$ の全てが端数部分 $M_F$ で、かつ、定義により入力信号サンプル系列Xのサンプルの仮数部23ビットが $a110$ である。 $-126$   $E - E_0 = -1$ の場合は仮数部が $a110$ ではない端数部分であり、いずれにしても $E - E_0$ が負の場合は整数値信号サンプル系列Yのサンプルの全ビットは $a110$ となり、差分信号Zは入力サンプル系列Xのサンプルと同じになるので、浮動小数点形式の信号サンプル系列Xのサンプルの全32ビットを全て符号化する。

10

【0027】

整数値の絶対値 $X_w$ が1の場合は、仮数部Mの23ビット目より1ビット上位のビット(以下24ビット目という)  $1$ により整数値 $X_w$ を表し、仮数部Mの23ビット $x_{23} \dots x_1$ は端数部分 $M_F$ を表わす。

整数値の絶対値 $X_w$ が2~3の場合は24ビット目の  $1$ と23ビット目の $x_{23}$ との2ビット  $1x_{23}$ により整数値 $X_w$ を表わし、 $x_{23}$ は仮数部M中の整数部分 $M_I$ を表し、 $x_{22} \dots x_1$ の22ビットで端数部分 $M_F$ を表わす。

$X_w$ が4~7の場合は24ビット目の  $1$ と23ビット目 $x_{23}$ 、22ビット目 $x_{22}$ との3ビット  $1x_{23}x_{22}$ により整数値 $X_w$ を表わし、 $x_{23}x_{22}$ は仮数部M中の整数部分 $M_I$ を表し、 $x_{21} \dots x_1$ の21ビットにより端数部分 $M_F$ を表わしている。

20

【0028】

$X_w$ が $2^{k-1} \sim (2^k - 1)$ の場合は、24ビット目の  $1$ と $x_{23} \dots x_{23-(k-2)}$ とのkビットで整数値 $X_w$ を表わし、 $x_{23}x_{22} \dots x_{23-(k-2)}$ は仮数部M中の整数部分 $M_I$ を表し、 $x_{23-(k-1)} \dots x_1$ の $(23-k)$ ビットで端数部分 $M_F$ を表わす。

所で差分信号Zは32ビット浮動小数点形式の入力サンプル系列Xから、整数値信号サンプル系列Yを32ビット浮動小数点形式にしたサンプル系列Yを差し引いたものであるから、差分信号Zは、図5に示すようになる。整数値の絶対値 $X_w$ が0の場合は前述の $E - E_0$ が負の場合であり、差分信号Zの極性S、指数部E、仮数部Mは全て入力サンプルXと同一になる。

30

【0029】

整数値の絶対値 $X_w$ が1の場合は、サンプル系列YのサンプルのS、Eはそれぞれサンプル系列Xのサンプルのそれらと同一となりMは0となる。ここでは指数部の差分もとる。従って差分信号ZのS、Eはそれぞれ必ずXと同じ0、0となり、Mはサンプル系列Xのサンプルの仮数部Mと同一となる。

整数値 $X_w$ が2~3の場合は、サンプルYのS、Eはそれぞれ、サンプルXのそれと同一となり、Mは23ビット目 $x_{23}$ がサンプルXのMのそれと同一となり、端数部分 $x_{22} \dots x_1$ は全て0である。従って差分信号ZのS、Eは必ず0となり、仮数部Mは上位1桁(23ビット目)は必ず0となり、残りの端数部分 $x_{22} \dots x_1$ がサンプルXの対応する部分の値と同じであり、サンプルXにより変化する。

40

【0030】

以下同様に、サンプルXの整数値 $X_w$ が $2^{k-1} \sim (2^k - 1)$ の場合、差分信号ZのS、Eは必ず0となり、仮数部Mは、サンプルXの仮数部Mの整数部分と対応する上位 $(k-1)$ 桁(ビット) $x_{23} \dots x_{23-(k-2)}$ は必ず0となり、残りの $23-(k-1)$ ビット $x_{23-(k-1)} \dots x_1$ はサンプルXの仮数部Mのそれと同一値となり、サンプルXにより変化する。このように、整数形式の信号サンプル系列Yの整数値の範囲、つまり整数値の桁数 $n+1$ に応じて差分信号Zの非ゼロになり得る桁数 $h = 23 - n$ が決まる。

50

このような関係があるから、浮動小数点形式差分信号Zについては、圧縮部17において、整数化部12から入力された仮数部の整数部分の桁数nを用い、対応差分信号Z中の非ゼロになり得る下位 $h = (2^3 - n)$ 桁についてのみ可逆圧縮符号化する。つまり仮数部の整数部分の桁数nが0であれば指数部E、仮数部Mの全てを可逆圧縮符号化し、nが0であれば、仮数部M中の下位 $h = (2^3 - n)$ ビットのみを可逆圧縮符号化する。以上から入力信号サンプルXと仮数部の整数部分の桁数nとを差分生成部14に入力して、差分信号Zを生成できることが理解されよう。

#### 【0031】

圧縮部17による差分信号Zの符号化は、前述のように入力信号サンプル系列Xのサンプルの $E - E_0$ が $-127 \sim 0$ の場合は、差分信号Zは入力信号サンプルXと同一になり、従って入力信号サンプルXの全32ビットを差分信号Zとして符号化する。 $E - E_0$ が $1 \sim 22$ の場合は、例えば差分信号サンプル系列 $Z = z_1, z_2, z_3 \dots$ が図6に示す状態の場合、nが0の場合以外は非ゼロになり得る桁部分 $M_D$ のみを符号化する。この非ゼロになり得る桁についての符号化はその非ゼロになり得る桁数h毎に例えばハフマン符号化などのエントロピー符号化を行うか、まとめて例えばユニバーサル符号化を行う。つまり図6中に斜線を施した部分のみをサンプル毎にあるいはフレーム単位など適宜にまとめてエントロピー符号化などの可逆圧縮符号化を行えばよい。各サンプルの非ゼロとなり得る桁部分 $M_D$ の桁数(ビット数)hは $h = (2^3 - n)$ と求まるので、フレーム毎に全ての $M_D$ を一連のビット列として符号化しても、復号側で復号化された一連のビット列を各サンプルの非ゼロとなり得るビット数hごとに分離可能である。

#### 【0032】

なお、極性Sは整数形式の信号サンプル系列Yの符号化により、符号Caに含まれているため、差分信号Zの符号化の際に極性Sを符号化して送信する必要はない。また、受信側では整数値信号サンプル系列Yの各サンプルの極性を除く最上位の1より下位のビット数nから $n = E - E_0$ を求めることができるので、桁数nを送信する必要はない。

差分信号Zは小数点以下の桁数が多いこともあり、その場合、符号化の効率が悪い場合がある。従って、図1の実施形態において差分信号Zの非ゼロとなり得る桁を符号化した場合に得られる符号列の情報量と、符号化しない場合の非ゼロになり得る桁の情報量を比較し、後者のほうが情報量が小さければ非ゼロとなり得る桁を符号化せずそのまま符号列Cbとして出力してもよい。あるいは、この差分信号Zのサンプルの全桁を符号化した場合としない場合の情報量を比較し、情報量が小さくなるほうを選択してもよい。

#### 【0033】

なお前述したように浮動小数点形式の信号サンプル系列X自体の仮数部が $E = 150$ の例外的な場合は、整数化部12で指数部Eが150に固定されたことを示す例外信号yが差分生成部14に与えられ、差分生成部14はその指数部Eの値150と信号サンプル系列Xのサンプルの指数部Eとの差分( $E - 150$ )を指数部とし、仮数部Mの差分を仮数部とする浮動小数点形式の差分信号Zを生成する。圧縮部17は差分信号Zを可逆圧縮符号化し、その符号列Cbとして出力する。

図1に示すように、1系統符号化部120では、入力された浮動小数点形式の入力信号サンプル系列Xは圧縮部121で直接可逆圧縮符号化され、符号列Ccとして出力される。この可逆圧縮符号化は例えば8ビット単位のユニバーサル符号化で行えばよい。分析選択部100より2系統符号化部110と1系統符号化部120とのいずれを選択したかを示す補助情報が補助符号化部103により補助符号Cdとして符号化されて出力される。これら符号列Ca、Cbと補助符号Cd又は符号列Ccと補助符号Cdが出力部104から出力され、伝送、あるいは記録媒体への記録に用いられる。

#### 【0034】

図7に図1に示した符号化装置と対応するこの発明による復号化装置の実施形態を示す。

入力端子201よりの符号列は1フレーム分の符号列毎に補助符号抽出部202で補助符号Cdが抽出され、補助符号Cdにより分離部203が制御され、入力端子201より

の符号列は補助符号 C d が 2 系統符号化を表わす場合は入力端子からの符号列 C a と C b が分離されて 2 系統復号部 2 1 0 へ供給され、1 系統符号化を表わす場合は入力端子からの符号列から符号列 C c が分離されて 1 系統復号部 2 2 0 へ供給される。

#### 【 0 0 3 5 】

2 系統復号部 2 1 0 に入力された符号列 C a は伸張部 2 1 で可逆伸張復号化される。この可逆伸張復号化方法は、図 1 中の圧縮部 1 3 で行った可逆圧縮符号化方法と対応し、その処理と逆の処理を行う。よってこの可逆伸張復号化により 1 サンプル 2 4 ビットの整数形式のデジタル信号系列 Y が生成される。また伸張部 2 1 中の桁数出力部 2 1 A において、伸張復号された各サンプルごとの最上位の " 1 " より下位の桁数 n を浮動小数点形式のデジタル信号の仮数部 M における整数部分の桁数として求めて出力される。この桁数 n は伸張部 2 3 に入力される。

10

#### 【 0 0 3 6 】

伸張部 2 3 ではこれに入力された符号列 C b が可逆伸張復号化される。この可逆伸張復号化方法は図 1 中の圧縮部 1 7 で行う可逆圧縮方法と対応したものとする。よってこの伸張部 2 3 において、伸張復号されたビット列から、組立部 2 3 A により桁数 n に基づいて、サンプルごとの 3 2 ビット浮動小数点形式の差分（誤差）デジタル信号が組立てられ、浮動小数点形式の差分信号 Z として出力される。

組立部 2 3 A では、伸張部 2 1 からのフレームの最初のサンプルに対する桁数 n と、伸張部 2 3 で伸張復号されたビット列から、最初のサンプルと対応する部分として  $n = 0$  でなければ、2 3 ビットの仮数部 M の下位  $2 3 - n = h$  ビットとして h ビットが取り出されて、図 6 中の最初のサンプル  $z_1$  における斜線を施した部分よりなる 1 サンプル 3 2 ビットの浮動小数点形式の差分信号サンプル  $z_1$  として組立てられる。極性 S の 1 ビット、指数部 E の 8 ビットはすべて 0 とし、また図 6 中の仮数部 M 中の整数値と対応する部分  $M_E$  は全て 0 とされる。次のサンプルに対して対応桁数  $n = 0$  に応じて仮数部の下位  $2 3 - n = h$  ビットとして伸張復号ビット列から h ビットを取り出して 3 2 ビットの浮動小数点形式差分信号 Z とし、以下同様にして図 6 に示したようなサンプル列  $z_2, z_3 \dots$  が順次組立てられて出力される。 $n = 0$  の場合は指数部 E の 8 ビットと仮数部 M の 2 3 ビットが伸張復号ビット列から取り出されて浮動小数点形式デジタル信号とする。この復号した差分信号 Z は極性 S は常に 0、指数部 E は  $n = 0$  以外は常に 0、仮数部 M は下位の  $2 3 - n = h$  ビットのみが非ゼロになり得るビットであり、その他は常に 0 であり、従って厳密な意味では浮動小数点表現ではないが、ここでは浮動小数点形式差分信号と呼んでいる。

20

30

#### 【 0 0 3 7 】

伸張部 2 1 よりの 2 4 ビット整数形式の信号サンプル系列 Y は浮動小数点化部 2 2 で 1 サンプル 3 2 ビットの浮動小数点形式の信号サンプル系列  $Y'$  に変換される。この信号サンプル系列  $Y'$  は伸張部 2 3 よりの浮動小数点形式の差分信号 Z とが合成部 2 4 で対応サンプル毎に加算合成され、浮動小数点形式の信号サンプル系列 X が再生される。再生信号サンプル系列 X の各サンプルの極性は信号サンプル系列  $Y'$  のそれ、つまり復号された整数形式の信号サンプル系列 Y の極性となり、サンプル列 X のサンプルの指数部 E は  $n = 0$  以外は、信号サンプル系列  $Y'$  のサンプルの指数部となり、仮数部 M は上位  $n = 2 3 - h$  ビットが信号サンプル系列  $Y'$  のサンプルの仮数部 M の上位 n ビットとなり、下位  $2 3 - n = h$  ビットが差分信号 Z の下位 h ビットとなる。

40

#### 【 0 0 3 8 】

1 系統復号部 2 2 0 に入力された符号列 C c は、図 1 中の圧縮部 1 2 1 と対応した可逆復号伸張処理により、原浮動小数点形式の信号サンプル系列 X に直接復号される。

2 系統復号部 2 1 0 及び 1 系統復号部 2 2 0 で復号された浮動小数点形式の信号サンプル系列 X はフレーム合成部 2 0 5 において、順次連続合成されて、浮動小数点形式信号サンプル系列とされる。

#### 変形実施形態

図 8 は図 1 で示した符号化装置の変形実施形態を示す。この符号化装置は、図 1 における分析選択部 1 0 0 を使用しない。ここでは 2 系統符号化部 1 1 0 及び 1 系統符号化部 1

50

20で入力信号サンプル系列Xをそれぞれ一旦符号化し、分析選択部100に符号列Ca及びCb、また符号列Ccを入力し、そのフレームにおける圧縮効率を効率計算部100cでそれぞれ計算する。例えばそれぞれの符号長のそのフレーム内での総和を求め、比較判定部100dで比較し、その符号長総和が小さい方が、圧縮効率が高いものとして、判定結果を補助符号化部103に与えて符号Cdを生成するとともに、比較結果により出力部104を制御して出力部104から、符号列の組(Ca, Cb, Cd)又は符号列の組(Cc, Cd)を選択出力する。その他の構成は図1の場合と同様であり、説明を省略する。図中に破線で示すように、差分信号Zをそのまま出力する場合も選択の1つとして加え、符号列の組(Ca, Cb, Cd), (Ca, Z, Cd), (Cc, Cd)のいずれか最も圧縮効率の高いものを出力してもよい。図8の符号化装置に対応する復号化装置としては、図7に示した復号化装置をそのまま使用することができる。

10

#### 【0039】

なお図1に示した符号化装置において、圧縮部17は浮動小数点形式の差分信号Zを直接可逆圧縮符号化した場合は、図7に示した復号装置中の伸張部23ではnを使用せず、浮動小数点形式の差分信号Zが直接復号されるようにする。この直接可逆圧縮符号化、復号化する場合は整数化部12における整数化は切り捨てのみならず、四捨五入、切り上げにより行ってもよい。

#### 変形実施形態

一般の浮動小数点ではなく、もともと24ビットとか16ビットの整数値のPCM信号を便宜上、浮動小数点に変換して符号化対象の系列を作成する場合がある。このような特殊な浮動小数点系列の場合には、前述した第1実施形態において非0になり得る桁はなくなり、差分生成部14よりの浮動小数点形式の差分(誤差)信号Zの極性S、指数部E、仮数部Mはすべてゼロとなり、送信する必要はない。この場合は、例えば補助情報として2ビットを用い、その2ビットが00であればもともとのデジタル信号が1サンプル16ビットの整数値のみ、01であれば1サンプル24ビットの整数値のみ、10であればそれ以外の信号であることを復号化装置に伝えればよく、そのためには、この2ビット補助情報を例えば圧縮部17の出力符号列Cbの先頭に追加し、前2者の場合は符号列Cbとしてこの2ビットの補助符号のみを出力し、後者の場合には補助符号に符号列Cbを付加して出力することにより効率的な圧縮が可能となる。

20

#### 第2実施形態

30

この発明の第2実施形態では浮動小数点形式の信号サンプル系列Xから、必ず整数形式の信号サンプル系列Yを生成するが、浮動小数点形式の差分信号Zについては、2つの可逆圧縮符号化方法のいずれかの圧縮効率が高いと推定される方により符号化する。つまり1つは第1実施形態で説明したように整数形式信号サンプル系列Yの各サンプルの桁数を利用して浮動小数点形式の差分信号の非ゼロとなり得る桁のみを可逆符号化する方法であり、他の1つは浮動小数点形式の差分信号を直接可逆圧縮符号化する方法である。

#### 【0040】

この第2実施形態の符号化装置の機能構成を、図9に、図1と対応する部分に同一参照番号を付けて示し、重複説明は省略する(以下も同様)。この第2実施形態では差分信号Zの符号化のために、整数値サンプルの最上位の1より下位の桁数nを利用する圧縮部17と、浮動小数点形式の差分信号を直接可逆圧縮符号化する圧縮部131とが設けられる。また分析部132に整数化部12から仮数部M中の整数部分の桁数(整数値サンプルの最上位の"1"より下位の桁数)nが入力され、その桁数nと基準値とが判定部132aで比較され、その比較結果に応じて選択部133が制御される。差分信号Zの仮数部Mの上位nビットは"0"であり、nが大きいほど、圧縮部17を使用するほうが圧縮効率が高くなる。そこで、桁数nが基準値以上であれば、差分信号Zを圧縮部17へ供給し、桁数nが基準値以上でなければ差分信号Zを圧縮部131へ供給する。圧縮部17と131のいずれかで可逆圧縮符号化された差分信号Zに対する符号列Cbが出力部104に入力されることになる。

40

#### 【0041】

50

桁数  $n$  が大きければ、つまり振幅が大きければ、図 5 を参照して説明したように、差分信号  $Z$  の各サンプル中の非ゼロをとり得る桁の数  $h = 2^3 - n$  が小さくなり、差分信号  $Z$  を効率的に圧縮することができる。しかし桁数  $n$  が小さければ、つまり振幅が小さければ、差分信号  $Z$  中の非ゼロを取り得る桁数  $h$  が大きくなり、圧縮部 1 3 1 において、浮動小数点形式の差分信号  $Z$  を、例えば通常の 8 ビット単位のユニバーサル符号化により可逆圧縮符号化をした方が有利である。例えば、各差分信号サンプルの 2 3 ビットの仮数部の先頭（又は末尾）に 0 を 1 ビット挿入して 2 4 ビットのビット列とし、そのビット列を 8 ビット毎の 3 個のシンボル（符号化単位）に分解し、8 ビットのシンボル毎の圧縮符号化を適用する。あるいは、各差分信号サンプルの仮数部の非ゼロとなり得るビットのみを 8 ビットごとに区切り、8 ビットで割り切れないものには 0 を必要なビット数だけ追加挿入して、8 ビット毎のシンボルを作成し、シンボル毎に圧縮符号化を適用してもよい。例えば、あるサンプルでは仮数部の非ゼロとなり得るビット数が 1 1 であれば、5 個の 0 を追加して 8 ビットの 2 つのシンボルを作る。

#### 【0 0 4 2】

判定部 1 3 2 a における前記基準値としては、例えば整数形式デジタル信号サンプル系列  $Y$  の振幅長が 1 6 ビットの場合、例えば 1 0 ビットとされる。この基準値も第 1 実施形態の判定部 1 0 1 b における所定値と同様に各種条件に応じて高い圧縮効率を得られるように選定される。

この圧縮部 1 7 と 1 3 1 との選択は、各サンプル毎に行ってもよく、あるいは、分析部 1 3 2 の平均部 1 3 2 b で 1 フレーム分の整数形式信号サンプルのそれぞれの最上位の " 1 " より下位の桁数  $n$  を平均し、その平均値が基準値以上であれば、圧縮部 1 7 を選択し、基準値以上でなければ圧縮部 1 3 1 を選択するようにフレーム毎に制御してもよい。各サンプル毎に圧縮部 1 7 と 1 3 1 との選択を行う場合は、圧縮部 1 7 では例えば入力された 1 フレーム分の選択されたそれぞれのサンプルの非ゼロとなり得るビットをビット連結して、エントロピー符号化する。桁数  $n$  は復号側において復号化された整数形式の信号サンプルから求めることができるので、桁数  $n$  を利用して圧縮部 1 7 と 1 3 1 の選択を行う場合は、その圧縮部 1 7 と 1 3 1 のいずれを選択したか表わす補助符号を出力する必要はない。

#### 【0 0 4 3】

図 9 に示した符号化装置と対応する復号装置の実施形態を図 1 0 に図 7 と対応する部分に同一参照番号を付けて示す。分離部 2 0 3 により分離された符号列  $C a$  は必ず伸張部 2 1 に入力されて、整数形式の信号サンプル系列  $Y$  が復号伸張される。符号列  $C b$  に対する伸張部として伸張部 2 3 の他に伸張部 2 3 1 が設けられる。

伸張部 2 1 よりの整数形式デジタル信号サンプルの最上位の " 1 " より下位の桁数  $n$  が分析部 2 3 2 に入力される。分析部 2 3 2 は図 9 中の分析部 1 3 2 と同一構成であり、桁数  $n$  と基準値とが判定部 2 3 2 a で比較され、桁数  $n$  が基準値以上であれば、符号列  $C b$  は伸張部 2 3 へ供給され、桁数  $n$  が基準値以上でなければ、符号列  $C b$  は伸張部 2 3 1 へ供給されるように選択部 2 3 3 が制御される。伸張部 2 3 では第 1 実施形態で説明したように、復号伸張されたビット列から桁数  $n$  で決まる非ゼロを取り得る桁数  $h$  ずつ取り出され、組立部 2 3 a で浮動小数点形式の差分信号  $Z$  に組み立てられる。

#### 【0 0 4 4】

伸張部 2 3 1 では、図 9 中の圧縮部 1 3 1 と対応した復号伸張方法により、符号列  $C b$  が浮動小数点形式の差分信号  $Z$  に直接復号伸張される。なお図 9 中の分析部 1 3 2 で桁数  $n$  の平均を利用した場合は分析部 2 3 2 においても桁数  $n$  を平均部 2 3 2 b で平均して、基準値と比較する。この比較結果により選択部 2 3 3 が制御され、符号列  $C b$  が伸張部 2 3 又は 2 3 1 に与えられる。伸張部 2 3 では、符号列  $C b$  を図 7 で説明したと同様に復号して非ゼロとなり得る桁のビット列を得る。そのビット列から桁数  $n$  を使って差分信号の仮数部を形成し、更に、浮動小数点形式の差分信号  $Z$  を組み立てる。

#### 【0 0 4 5】

前述したように、ここでは差分信号  $Z$  は浮動小数点形式とは言っても符号化装置の差分



生成部 1 4 において生成された差分の指数部と極性はともに全て " 0 " となるため送信されず、復号化装置の伸張部 2 3 で復号した非ゼロとなり得る桁から仮数部を組み立て、全てゼロの指数部とゼロの極性を付加して浮動小数点形式の差分信号 Z を組み立てている。

伸張部 2 3 1 は符号列 C b が符号列 C b ' として与えられ、図 9 における圧縮部 1 3 1 と逆の復号化処理を行い、直接差分信号 Z を得る。これらのうちのいずれかの差分信号 Z は加算部 2 4 で整数形式のサンプル Y に対応する符号小数点形式のサンプル Y ' が加算され、現浮動小数点形式の信号サンプル X が得られる。

#### 変形実施形態

図 1 1 は図 9 の実施形態の変形例であり、図 9 に対応する部分には同じ参照番号が付けられている。この変形実施形態では、図 9 において、分析部 1 3 2 と選択部 1 3 3 を使用する代わりに、圧縮部 1 7、1 3 1 で圧縮符号化を行って、どちらが圧縮効率の高い符号化が可能であるか判定し、効率の高いほうの符号化結果を出力する。即ち、差分信号 Z を圧縮部 1 7 と圧縮部 1 3 1 との両方でそれぞれ可逆圧縮符号化し、これらの符号化結果としての符号列 C b と C b ' をそれぞれ分析選択部 1 3 4 に入力して、そのフレームにおける差分信号 Z に対する圧縮効率が圧縮部 1 7 と 1 3 1 とでいずれが高いかを効率計算部 1 3 4 a 及び判定部 1 3 4 b により判定して、その圧縮効率が高い方の符号列を符号列 C b として出力部 1 0 4 より出力してもよい。この場合は、符号列 C b が圧縮部 1 7 と 1 3 1 のいずれにより符号化されたものであるかを表わす補助符号 C e を補助符号化部 1 0 3 から出力する必要がある。

#### 【 0 0 4 6 】

図 1 1 の符号化装置に対応する復号化装置は、図 1 0 に示した復号化装置において、分析部 2 3 2 を設ける代わりに破線で示す補助復号化部 2 3 4 を設けた構成とする。分離部 2 0 3 により分離された補助符号 C e を補助復号化部 2 3 4 で復号し、その復号結果により選択部 2 3 3 を制御して符号列 C b を伸張部 2 3 又は 2 3 1 に与える。その他の動作は図 9 の符号化装置に対する復号化装置としての動作と同様である。

図 9 の構成に図 1 1 における分析選択部 1 3 4 と補助符号化部 1 0 3 を設け、桁数 n を使った判定と圧縮効率を使った判定のいずれを使用するかを選択できるようにし、図 1 0 においても分析部 2 3 2 に加えて補助復号化部 2 3 4 を設けてどちらの手法でも使用できるようにしてもよい。ただし、符号化側と復号化側で対応する手法を使用する。

#### 第 3 実施形態

この発明の第 3 実施形態は図 1 の第 1 実施形態と図 9 の第 2 実施形態とを組み合わせたものである。この第 3 実施形態の符号化装置の機能構成図を図 1 2 に、図 1、8、9 及び 1 1 と対応する部分に同一参照番号を付けて示し、重複説明は省略する。この実施形態によれば、選択部 1 0 2 と 1 3 3 の選択の組み合わせにより符号列の組 ( C a , C b , C d ) , ( C a , C b ' , C d ) , ( C c , C d ) のいずれかが選択され出力される。

#### 【 0 0 4 7 】

これらと同様の 3 つの組の符号列の選択は、図 1 1 の実施形態に破線で示すように図 8 の圧縮部 1 2 1 を追加し、最も圧縮効率の高い符号列の組を分析選択部 1 3 4 で選択することにより実現できる。

この第 3 実施形態の復号装置を、図 1 3 に、図 7 及び図 1 0 と対応する部分に同一参照番号を付けて示し、重複説明を省略する。

#### 第 4 実施形態

上述の第 1 実施形態では、浮動小数点形式の入力信号サンプル X の小数点以下の端数を切り捨てることにより整数化し、差分信号 Z の仮数部における整数部分が a 1 1 0 となることを利用して差分信号 Z の非ゼロとなり得る桁のみを符号化することにより圧縮効率を高める符号化装置を示した。また、第 2 実施形態では、差分信号 Z について、非ゼロとなり得る桁のみを符号化する場合と全桁の符号化する場合とで効率の良い方を選択する符号化装置を示した。第 4 実施形態では、上記小数点以下の切捨てに加えて、更に整数部分も予め決めたビット数を切り捨てて整数値信号サンプル Y を生成する例を示す。以下の例では、小数点以下の切捨てによる 1 6 ビットの整数値化において、更に 8 ビットの切捨

てを行う場合を示すが、17～23ビットの任意の整数値化にも適用できる。

#### 【0048】

この第4実施形態によれば浮動小数点形式の入力信号サンプルXが、もともと符号小数点信号であったのか、24ビット整数値信号であったのか、16ビット整数値信号であったのかを区別する必要がなく、同じ処理によって符号化を行うことができるが、動作原理の説明上、浮動小数点形式の入力信号サンプル系列Xがもともと浮動小数点形式であった場合、極性1ビットを含む24ビット整数値信号であった場合、極性1ビットを含む16ビット整数値信号であった場合について順に説明する。

図14Aを参照して、もともと浮動小数点形式であった入力信号サンプルXの16ビット整数値化と、差分信号Zの生成を説明する。図において  $x$  は 0 又は 1 のいずれかをとり得る（非ゼロとなり得る）。 $M_x$  は浮動小数点形式の入力信号サンプルXの仮数部23ビットを示し、上位  $n (= E - E_0)$  ビットの整数部分と小数点以下の端数部分の境界を示す小数点位置を  $P_D$  で示してある。サンプルXの指数部Eを  $E - 8$  に削減することにより仮数部  $M_x$  の小数点位置  $P_D$  は8ビット上位にシフトする。その結果、図14Aの例では仮数部  $M_x$  の上位  $n - 8 = 2$  ビット  $xx$  がシフト後の仮数部内の整数部分となっている。その整数部分2ビットの隣接上位側に 1 を加えてた  $1xx$  を16ビット整数値信号サンプルYの最下位ビットとする。

#### 【0049】

16ビット整数値信号サンプル系列Yのサンプルを8ビット上位にシフトし、即ち、最下位に8ビットの 0 を付加してから浮動小数点形式の信号サンプル系列Y'に変換する。従って、得られた浮動小数点形式の信号サンプル系列Y'の指数部Eは8を加算されたことになり、入力信号サンプル系列Xの指数部Eと同じ値になる。信号サンプル系列Y'の仮数部  $M_{y'}$  は、8ビット上位にシフトされた整数値信号サンプル系列Yの最上位の 1 より下位の整数部分、ここでは2ビットの  $xx$  を最上位ビットとし、それより下位は全て 0 とすることにより得られる。図14Aでは便宜上8ビットシフトする前のサンプル系列Yの最上位の 1 より下位の整数部  $xx$  と浮動小数点形式の信号サンプル系列Y'に変換後の仮数部  $M_{y'}$  の関係を示している。差分信号Zの仮数部  $M_z$  は仮数部  $M_x$  と  $M_{y'}$  の差分であり、Zの仮数部  $M_z$  の上位  $(n - 8)$  ビット、ここでは最上位2ビット、は 00 となり、それより下位（端数部分）は仮数部  $M_x$  の対応するビット位置の値と同じである。即ち、仮数部  $M_z$  の最上位の整数部分 00 に続く下位の全ビットは非ゼロとなり得る。

#### 【0050】

差分信号Zについては仮数部Mのこの非ゼロとなり得る下位  $23 - (n - 8)$  ビットのみを符号化すればよい。仮数部  $M_z$  の必ず0となる上位  $(n - 8)$  ビットは、受信側で復号した整数値信号サンプル系列Yの最上位の 1 より下位の全ビット数と同じである。この仮数部Mの非ゼロとなり得る領域は図15Aに示すように、整数部分のビット数  $(n - 8)$  が増加すると共に減少する。

図14Bは24ビット整数値信号から浮動小数点形式の入力信号サンプル系列Xが生成された場合であり、従って、サンプル系列Xの仮数部  $M_x$  は図14Bに示すように小数点  $P_D$  より下位で全ての桁が 0 となっている点が図14Aの仮数部  $M_x$  と異なる。図14Bにおいても入力信号サンプルXの指数部Eを  $E - 8$  にすることにより小数点  $P_D$  の位置が8ビット上位側にシフトするが、その結果から生成される16ビット整数値信号サンプル系列Yは図14Aの場合と同じになる。従って、整数値信号サンプル系列Yを浮動小数点形式に変換したときの仮数部  $M_{y'}$  も図14Aの場合と同様である。

#### 【0051】

しかしながら、差分信号の仮数部  $M_z$  は仮数部  $M_x$  と  $M_{y'}$  の差分であり、図14Bに示すように差分信号Zの仮数部  $M_z$  の整数部分は 00 となり、それより下位8ビットが非ゼロとなり得るビット範囲であり、更にその下位側は全てのビットが 0 となる。この非ゼロとなり得るビットの範囲は整数部分のビット数が増加するにつれ図15Bに示すように下位側にずれていく。

図14Cは極性1ビットを含む16ビット整数値信号サンプルから浮動小数点形式の入力信号サンプルが作られている場合である。従って、23ビットの仮数部 $M_x$ において、小数点 $P_0$ と隣接する下位側8ビットは全て0であり、それより更に下位の桁も全て0である。小数点以下の桁数 $h$ は $h = 23 - n$ と $n$ により変化するが、8より小さくはない点図14Bの場合と異なる。

【0052】

図14Cにおいても入力信号サンプル系列 $X$ の指数部 $E$ を $E - 8$ にすることにより小数点 $P_D$ の位置が8ビット上位側にシフトし、シフト後の整数部分 $x_x$ から16ビット整数値信号サンプル系列 $Y$ が図14A, 14Bと同様に生成される。従って、サンプル系列 $Y$ を浮動小数点形式に変換したときの仮数部 $M_y$ も図14A, 14Bの場合と同じである。また、差分信号 $Z$ の仮数部 $M_z$ は図14Bの場合と同様にその最上位( $n - 8$ )桁の整数部分が0となり、それに続く下位8ビットは非ゼロを取りえる範囲であり、その更に下位ビットは全て0となる。この場合も整数部分の桁数( $n - 8$ )が増加すると図15Bに示すように非ゼロとなり得る8ビットの下位側の0の桁数も減っていくが、8桁より小さくなることはない。

【0053】

以上の説明から明らかなように、符号小数点形式の入力信号サンプル $X$ のものと信号が24ビット整数値信号、16ビット整数値信号、浮動小数点形式信号のいずれの場合であっても、8ビット切り捨てによる16ビット整数値信号サンプル $Y$ を生成するには、入力信号サンプル $X$ の指数部 $E$ を $E - 8$ として小数点位置を8ビット上位側にシフトし、仮数部 $M_x$ の整数部分である上位( $n - 8$ )ビット、図の例では $x_x$ 、に対しその上位側に1を加えた整数値 $1x_x$ を最下位ビットとする16ビット整数値信号サンプル系列 $Y$ を構成すればよい。また、差分信号 $Z$ の仮数部 $M_z$ は、入力信号サンプル $X$ の仮数部 $M_x$ の最上位( $n - 8$ )ビットを0とすることにより得られる。

【0054】

このようにして得られた16ビット整数値信号サンプル $Y$ に対しては、図1の実施形態と同様に圧縮符号化すればよい。一方、差分信号 $Z$ については、その仮数部 $M_z$ を、常に0である最上位( $n - 8$ )ビットについては符号化せず、それより下位の非ゼロとなり得る8ビットの領域と、更にそれより下位の残りの( $23 - n$ )ビットの領域を別々に圧縮符号化する。

上述の例では16ビット整数値とするため、23ビットの仮数部の小数点の位置を8ビット上位にシフトして、そのシフトした小数点以下を切り捨てる場合を示したが、一般に、 $22 - m - 8$ の任意の整数 $m$ ビットの切捨てを行うには、入力信号サンプルの指数部 $E$ を $E - m$ として仮数部 $M_x$ の上位( $n - m$ )ビットを整数部分とるように切捨てを行えばよい。また、差分信号 $Z$ の仮数部の圧縮符号化は、最上位( $n - m$ )ビットの0は符号化せず、その下位の $m$ ビットの非ゼロとなり得る領域と、更に下位の残りの( $23 - n$ )ビットの領域に分けて別々に符号化することにより効率のより符号化が可能となる。

【0055】

図16に符号化装置の例を図12と対応する部分に同一参照符号を付けて示す。選択部102からの浮動小数点形式の入力信号サンプル系列 $X$ は桁調整部31で図14A, 14B, 14Cで説明したように指数部 $E$ を $E - 8$ とすることにより小数点位置は上位側に8ビットシフトされる。整数化部12では8ビットシフトされた小数点以下の端数部分を切り捨てて、整数部分の上位側に1を付加して16( $= 24 - 8$ )ビットの整数形式の信号サンプル系列 $Y$ とする。この信号サンプル系列 $Y$ が圧縮部13で可逆圧縮され、符号 $C_a$ が出力される。

【0056】

またこの16ビット整数形式の信号サンプル系列 $Y$ は逆桁調整部32で桁調整部31と同一ビット数8であるが逆方向に、つまり最下位に8ビットの0を挿入することにより上位に8ビットシフトして、24ビットの整数形式信号サンプルとされる。この24ビット整数形式に変換された信号サンプルは浮動小数点化部15で浮動小数点形式の信号サ

サンプル系列 $Y$ に変換される。この浮動小数点形式信号サンプル系列 $Y$ と入力信号サンプル系列 $X$ との差分が減算部16でとられ、浮動小数点形式の差分信号 $Z$ が生成される。現実的にはこのような逆桁調整部32、浮動小数点化部15及び減算部16の処理を行うことなく、差分生成部14で整数化部12よりの仮数部の整数部分の桁数 $n$ を用いて、 $n=0$ で各入力信号サンプル $X$ の仮数部 $M_x$ の下位 $8+h(=15-n)$ ビットを取出し、 $n=0$ では指数部 $E$ と仮数部 $M_x$ とを取出して差分信号 $Z$ とすればよい。差分信号 $Z$ は図12の例と同様に、分析部132の出力に応じて圧縮部17又は131に与えられる。

#### 【0057】

圧縮部17は分配部33と2つの圧縮部17h、17cとから構成されている。この差分信号 $Z$ は分配部33に入力され、分配部33には整数化部12よりシフト後の各サンプル $X$ の仮数部の整数部分の桁数 $n$ が入力され、差分信号 $Z$ の各サンプルは、 $n=0$ の場合、桁数 $n$ に基づく非ゼロになり得る桁部分 $Z_h$ 、つまり下位の $h=15-n$ ビットと、整数形式信号サンプル系列 $Y$ を1サンプル24ビットから1サンプル16ビットに変更したことに基づく、つまり整数形式信号サンプル $Y$ のビット数に基づく、非ゼロになり得る桁部分 $Z_c$ 、つまり下位第 $(h+1)$ ビット～第 $(h-8)$ ビットとに分配する。 $n=0$ の場合、指数部 $E$ の8ビットと仮数部の下位 $23-8$ ビットを桁部分 $Z_h$ として、上位8ビットを桁部分 $Z_c$ として分配する。桁部分 $Z_h$ は圧縮部17hで、桁部分 $Z_c$ は圧縮部17cでそれぞれエントロピー符号化などにより可逆圧縮符号化され、符号列 $C_{b1}$ 、 $C_{b2}$ として出力される。

#### 【0058】

従って圧縮部17hには、主として仮数部 $M$ における下位の非ゼロになり得るビットが入力され、これが可逆圧縮符号化される。もともとのデジタル信号が24ビット又は16ビットの整数値であれば図14B、14Cに示したように $n=0$ 以外は圧縮部17hに入力される下位 $h$ 桁部分 $Z_h$ は全て0になるから、効率よく圧縮される。

圧縮部17cでは、図14A、14B、14Cに示したように、入力信号サンプル $X$ がもともと一般浮動小数点の場合、24ビット整数値の場合、16ビット整数値の場合に関係なく、非ゼロになり得る下位第 $(h+1)$ ～第 $(h+8)$ ビットの8ビットが可逆圧縮符号化される。このようにすることにより、図1中の圧縮部17で差分信号 $Z$ 中の非ゼロになりうるビットのすべてをまとめて圧縮符号化する場合より全体として効率よい圧縮が可能となる。図16から理解されるように、この圧縮部17hと17cを用いる手法は、図12に示した実施形態の符号化装置のみならず、図1、8、9、11にそれぞれ示した各実施形態の符号化装置のいずれにも適用できる。

#### 【0059】

この図16に示した符号化装置と対応する復号化装置の実施形態を図17に図13の復号装置と対応する部分に同一参照番号を付けて示す。補助符号抽出部202からの補助符号 $C_d$ により分離部203から符号列 $C_a$ 、 $C_{b1}$ 、 $C_{b2}$ が取り出された場合は、符号列 $C_a$ は伸張部21で可逆伸張復号されて1サンプル16ビット整数形式の信号サンプル系列 $Y$ が復号される。伸張部23は2つの伸張部23h、23cと合成部41とにより構成されており、符号列 $C_{b1}$ と $C_{b2}$ はそれぞれ伸張部23h、23cで可逆伸張復号化され、これらの復号されたビット列 $B_h$ 、 $B_c$ は合成部41に入力される。合成部41には伸張部21で復号された各サンプルごとの整数値の最上位の1より下位のビット数 $n$ も入力される。 $n=0$ の場合は伸張部23hの出力ビット列 $B_h$ から、 $h=15-n$ ビットが取出され、伸張部23cの出力ビット列 $B_c$ から8ビットが取出されこれらが仮数部 $M$ の23ビット中の下位 $h$ ビットと、その上位の8ビットとして合成される。以下同様に桁数 $n$ に応じ、ビット列 $B_h$ と $B_c$ からそれぞれ $h$ ビットと8ビットが取出されて合成される。 $n=0$ の場合は指数部 $E$ としての8ビットと $23-8=15$ ビットがビット列 $B_h$ から取り出され、ビット列 $B_c$ から8ビットが取り出され、この8ビットは前記15ビットの上位側に挿入され、23ビットの仮数部が生成される。

#### 【0060】

復号された16ビット整数形式信号サンプル $Y$ は桁調整部42で下位に8ビットの0

を挿入することで上位へ8ビットシフトされ、24ビット整数形式信号サンプルが得られる。その後、浮動小数点化部15で32ビット浮動小数点形式信号サンプル系列Yに変換される。この信号サンプル系列Yと差分信号Zとが合成部24で合成されて、浮動小数点形式の信号サンプル系列Xが再生される。補助符号Cdにより分離部203で符号列Ccが得られた場合は、符号列Ccを伸張部221で復号して直接信号サンプル系列Xが得られる。

#### 【0061】

上述において、整数形式信号サンプル系列Yを1サンプル16ビット、24ビットの場合としたが、24ビット以下の任意の数mとしてもよく、その数mに応じて整数形式信号サンプルYのサンプル当りのビット数変更に基づく、非ゼロになり得る桁数を決定すればよい。更に上述において、入力浮動小数点形式信号サンプルXを、例えば1024サンプルごとのフレーム、または連続する複数のサンプルごと分割し、その分割単位ごとに圧縮符号化処理してもよい。このように分割単位ごとに圧縮符号化する場合は、整数化するビット数を分割単位ごとに都合のよいように決め、そのビット数を指定する補助符号も出力するようにしてもよい。

#### 【0062】

図16に示す符号化装置の圧縮部17として図1で示した第1実施形態による圧縮部17を使用し、図17の復号化装置の伸張部23として図7における伸張部23を使用してもよい。即ち、その場合は符号化装置において16ビットの整数値化を行うが、差分信号Zは0となる領域と非ゼロとなり得る領域の2つに分けず、図6で説明したように、仮数部の下位 $h = 23 - n$ ビットのみ符号化を行う。従って復号化装置の伸張部23においては、符号Cbからhビットを復号化し、その上位にnビットの0を挿入して仮数部を生成する。

#### 第5実施形態

前述の図1、8、9、11及び12で説明した各実施形態において、圧縮部17による圧縮符号化は、差分信号Zを例えば1フレームごとに時間方向に抽出したビット列を可逆圧縮符号化するように構成してもよい。その符号化方法を図18を参照して以下に説明する。

#### 【0063】

図18では1フレームが1024サンプルにより構成された場合を示し、また各サンプル $i$  ( $i = 0 \sim 1023$ )に対応して入力信号サンプル系列Xのサンプルの仮数部Mの $n$  ( $= E - E_0$ )ビット整数部分の整数値と、 $h$  ( $= 23 - n$ )ビットの端数部分のビット列を示してある。第1実施形態で図6を参照して説明したように、差分信号Zの各サンプルの仮数部Mの整数部分に対応する上位nビットは全て0であり、端数部分である下位 $h = 23 - n$ ビットが非ゼロとなり得るビットである。図18の符号化方法では、差分信号の仮数部の下位hビットのみを、それらhビットの最上位ビットMSBをそろえて配列し、その配列における各サンプルの振幅方向の各ビット位置jのビットを時間方向に順次抽出して得たビット列 $E_j$ を符号化を行う。

#### 【0064】

サンプルによっては数部のビット長 $h = 23 - n$ は異なるので、振幅方向における同一ビット位置j ( $j = 0 \sim h_{max} - 1$ ;  $h_{max}$ はフレーム内のサンプルの最大端数部分ビット長)の時間方向のスキャンにおいて、ビットが存在しないサンプルはスキップする。スキップするか否かの判定は、サンプルiの振幅方向のビット位置jで、そのサンプルの端数部分ビット長hとjを比較し、 $h > j$ であればそのビット位置のビットを抽出し、 $h < j$ であればスキップして次のサンプル $i + 1$ を判定すればよい。

例えばMSB側から $j = 17$ のビット位置における時間方向のスキャンにおいて、端数部分のビット長 $h = 23 - 6 = 17$ のサンプル番号 $i = 0$ のサンプルは最下位にビットを有するが、端数部分のビット長 $23 - 8 = 15$ のサンプル番号 $i = 1$ のサンプルはMSB側から第15番目以降( $j = 14$ 以降)にビットを持たないのでスキップされる。これら第15番目の抽出されたビットを1フレーム分まとめてビット列 $E_{16}$ としてエントロピ

一符号化する。他のビット位置  $j$  についても同様である。エントロピー符号化として算術符号やゴロム符号を行うと階層化ができるので便利である。

【0065】

復号側においては、復号した各整数値信号サンプル  $Y$  の最上位の  $1$  より下位のビット数  $n$  から整数部分の桁数  $n$  がわかるので、フレーム内の各差分信号  $Z$  の仮数部における端数部分のビット長  $h = 23 - n$  がわかる。従って、復号した各ビット列  $E_j$  に対し、フレーム内のサンプル番号  $i$  を順次走査してサンプル番号毎に振幅方向のビット位置  $j$  を  $h$  と比較し、 $j > h$  ならビット列  $E_j$  中のビット値をビット位置  $(i, j)$  に割り当てることで図18の仮数部における端数部分を再構成することができる。この再構成された各サンプルの端数部分に対し、その上位側にサンプルに対応する  $n$  ビットの  $0$  を挿入することで差分信号の  $23$  ビットの仮数部  $M$  を再構成することができる。

10

【0066】

図19は図18の符号化方法を図1、8、9、11、12におけるの圧縮符号化部17に適用した場合の機能構成を示す。差分生成部14（例えば図1参照）からの差分信号  $Z$  の1フレーム分のサンプル  $z_0 \sim z_1 \dots z_3$  が入力されると共に、対応する整数部分桁数  $n_0 \sim n_1 \dots n_3$  が記憶部17Bに記憶される。制御部17Cは各サンプル  $i$  の整数部分桁数  $n_i$  に基づいて、入力差分信号サンプル  $z_i$  の仮数部の上位  $n$  ビットを除く  $h = 23 - n$  ビットの端数部分をバッファ17Aに取り込む。これにより図18に示した仮数部における端数部分の配列が得られる。次に、制御部17Cは振幅方向のビット位置  $j$  ( $= 0 \sim h_{max} - 1$ ) において時間方向にサンプルを走査し、 $h > j$  を満足するビットを抽出してビット列  $E_j$  を得て、圧縮部17Dに与える。バッファ17A、記憶部17B、制御部17Cはビット列生成手段を構成している。圧縮部17Dはビット列  $E_j$  を可逆圧縮符号化して符号  $Cb$  を出力する。

20

【0067】

図20は図19の圧縮符号化部17に対応する差分信号復号化手段としての図7、10、13における伸張部23の機能構成を示す。受信された符号  $Cb$  は復号化部23Dで復号されてビット列  $E_j$  が得られる。一方、桁計算部21A（例えば図7参照）からの整数桁数  $n$  の1フレーム分  $n_0 \sim n_1 \dots n_3$  が記憶部23Bに記憶され、制御部23Cはサンプル番号  $i$  の順に対応する端数桁数  $h_i = 23 - n_i$  から  $h_i > j$  を満足するビット位置  $(i, j)$  に対し、ビット列  $E_j$  から順に1つつビットを分配してバッファ23A内に配列記憶することを繰り返すことにより、図18の端数部分のビット配列が得られる。1フレームについての全ビット列  $E_j$ ,  $j = 0 \sim h_{max} - 1$  のビット配列が終了すると、制御部23Cはバッファ23A内の各サンプル  $i$  に対応した端数部分の上位側に  $n_i$  ビットの  $0$  を挿入して差分信号の仮数部を生成し、差分信号  $Z$  として合成部24（図7）に与える。バッファ23A、記憶部23B、制御部23Cは再構成手段を構成している。

30

【0068】

図18の差分信号の仮数部の符号化においては、各ビット位置  $j$  で時間方向にビットを集めてビット列  $E_j$  とし、符号化する場合を示したが、ビット位置の走査方法の他の例を以下に説明する。図21ではサンプルの振幅方向の予め決めた複数ビット幅に渡って、ビット位置  $j$  の最後のサンプルのビットからビット位置  $j + 1$  の最初のサンプルのビット位置に続けて走査してビットを集め、複数のビット列  $E_2, E_18, \dots$  を生成している。従って、得られるビット列  $E_j$  の数は端数部分の最大ビット数  $h_{max}$  より少なくなる。このビット走査においても、各サンプル  $i$  の  $h_i < j$  となるビット位置はスキップする。このようなビット走査を行っても、各サンプル  $i$  の端数部分桁数は  $h_i = 23 - n_i$  からわかるので、復号側においてはビット列  $E_j$  のビットを前述と同様に分配していくことにより端数部分のビット配列を再構成することができる。

40

【0069】

図22は更に他のビット走査の例を示す。図21ではビット位置  $j$  のフレームの終わりからビット位置  $j - 1$  のフレームの始めに連続させることを複数ビット幅繰り返して走査したのに対し、ここではその複数ビット幅を振幅方向に  $MSB$  側から走査し、各サンプル

50

$i$  のそのビット幅の終わりから次のサンプル  $i + 1$  のビット幅の始めに続けて操作することをフレームの最後のサンプルまで繰り返す。この例においても、条件  $h_i < j$  となるビット位置はスキップされる。

図 2 3 は図 1 8 のビット走査における例外動作を説明するためのものである。切捨てによって整数部分がゼロになってしまう場合、即ち  $E - E_0$  が負になる場合には指数部も含めて全 3 2 ビットを符号化して送る必要がある。そこで、図 2 3 に示すように、サンプル  $i = 3$  の整数部分がゼロであった場合（従って  $n = 0$ ）は、図 1 の圧縮部 1 7 においてこのサンプル  $i = 3$  を仮数部のビット走査から除外して別途符号化する。

【0070】

入力信号サンプル系列  $X$  が整数から変換された浮動小数点形式の信号サンプル系列であった場合の、元の整数の桁数と切り捨てた整数値の桁数が異なる場合、例えば図 1 4 B, 1 5 B で説明したように、2 4 ビット整数をそのまま浮動小数点とした信号サンプル系列を入力とし、1 6 ビットの整数に切り捨てて符号化した場合、差分信号サンプルの仮数部の端数部分の桁のうち、上位 8 桁は非ゼロとなり得るが、それ以外はゼロとなる。このような入力信号サンプル系列  $X$  を符号化する場合に、例えば図 1 3 の手法を使えば、ビット列  $E_0 \sim E_7$  は非ゼロとなり得るが、それより下位のビット列  $E_8 \sim E_{21}$  は全て 0 となるので、効率的な符号化が可能である。同様に、図 2 1、2 2 の手法を使っても効率的な符号化が可能なのは容易に理解される。

【0071】

例えばオーディオを信号の加工などで必要に応じて行われるように、2 4 ビットの整数値信号サンプルを 1 . 5 倍（2 進法で 1 . 1 倍）してから浮動小数点形式の信号に変換し、1 6 ビットの整数に切り捨てて符号化する場合、2 4 ビット整数値信号サンプルを 1 . 5 倍すれば少なくとも小数点以下第 1 位が非ゼロとなり得るビットになる。即ち、図 1 4 B に示すサンプル  $X$  の仮数部  $M_x$  の小数点  $P_0$  以下第 1 位が非ゼロとなりえる。そのため、差分信号の仮数部  $M_z$  の非ゼロとなり得る領域は下位側に 1 ビット増加して 9 ビットになるが、この場合でも、その 9 ビットより下位側は全て 0 となるので、図 1 8、2 1、2 2、2 3 の手法を使って効率的な符号化が可能となる。また、全てゼロでなくても非ゼロ（即ち 1）の頻度が桁ごとに偏る場合があるので、偏りに合わせて適応的に可逆符号化を適用することもできる。

【0072】

いずれの場合にも、差分信号の仮数部における端数部分の  $M S B$  側からの非ゼロとなり得る桁数がそろるので、効率的に圧縮符号化できる。

上述の図 1 8 ~ 2 3 の仮数部の符号化方法は図 1 に基づく第 1 実施形態に適用する例として説明したが、図 1 4 A ~ 1 7 で説明した 1 6 ビット整数値に切り捨て処理を行う場合にも適用できる。その場合は、図 1 6 の圧縮部 1 7 を図 1 9 のように構成し、図 1 7 の伸張部 2 3 を図 2 0 の伸張部のように構成すればよい。

また、端数部分の桁数が同じになるようにサンプルを並び替えてスキャンしてもよい。例えば桁数が 2 3 となるサンプルだけを集め、次に桁数が 2 2 となるサンプルを集め、順次桁数が小さくなるサンプルを集める。この並び替えの逆は追加情報がなくても一意に決まり、スキャンするサンプルの数の管理が容易になる。

【0073】

図 1 8 に示したビット配列のスキャンでは、フレーム毎に差分信号  $Z$  の各サンプル  $z$  の仮数部  $M$  中の非ゼロとなり得る  $h$  ビットを  $M S B$  側にそろえて時間軸方向に各ビット位置でスキャンしてビット列  $E_j$  を生成する例を示したが、図 2 4 に示すように、各サンプルの非ゼロとなり得る  $h$  ビットを  $L S B$  側にそろえて同様のスキャンを行ってもよい。この場合も、圧縮符号化部 1 7 としては、図 1 9 に示したものと同様に構成すればよく、単に制御部によるスキャンの制御を変えるだけである。また、復号化装置における伸張復号部 2 3 も図 2 0 で示したものと同様に構成できる。

## 第 6 実施形態

図 2 5 は前述の図 1、8、9、1 1、1 2 に示した符号化装置における圧縮部 1 7 の他

10

20

30

40

50

の実施形態を示す。この実施形態では、非ゼロとなり得る端数部分のLSBを同一ビット位置にそろえて整数値のサンプル列とし、サンプル列方向の冗長性を除いて圧縮符号化を行うものである。図25に示すように、圧縮符号化部17は、端数部分抽出部17Eと、フレーム内平均部17Fと、減算器17Gと、符号化部17D1、17D2とから構成されている。差分生成部14からの差分信号Zは端数部分抽出部17Eで各サンプルの仮数部Mから下位 $h = 23 - n$ ビットの端数部分を1つの整数値サンプルFとして抽出する。フレーム内平均部17Fは各フレーム毎にフレーム内の抽出された整数値サンプルの平均値 $F_{ave}$ を計算し、符号化部17D1で符号化して符号列Cb1を出力する。平均値 $F_{ave}$ と各整数値サンプルFとの差を誤差サンプルFとして減算器17Gで求め、その誤差サンプルFを符号化部17D2で予測符号化を行い、予測誤差を可逆圧縮符号化して符号列Cb2を出力する。予測パラメータは別途補助情報として送ってもよいし、符号化装置と復号化装置で共通に適応的に更新してもよい。

10

#### 【0074】

図26は図25に対応する復号化装置における復号伸張部23の構成例を示し、図7、10、13に示した復号化装置における伸張部23に適用できる。この実施形態では、伸張部23は伸張部23B、23Cと、加算部23Dと組立て部23Aとから構成されている。伸張部23Bは、受信した符号列Cb1を伸張復号化してフレームごとの平均値 $F_{ave}$ を出力する。伸張部23Cは受信した符号列Cb2を伸張復号化して予測誤差を求め、更に予測復号化を経て誤差サンプルFを出力する。誤差サンプルFは平均値 $F_{ave}$ と加算部23Dで加算され、整数値サンプルFが得られる。組立て部23Aはこの整数値サンプルFと、伸張部21の桁数計算部21Aからの桁数nとを使って差分信号Zを組み立てる。

20

#### 【0075】

この実施形態は図16及び17の実施形態にも適用でき、その場合は、図16及び17における圧縮符号化部17と伸張部23をそれぞれ図25及び26における圧縮符号化部17と伸張部23で置き換えればよい。

### 第7実施形態

図27は、図1、8、9、11及び12における圧縮部17に適用する第7実施形態の圧縮部の構成を示す。図18～23では非ゼロとなり得るビット列の配列をスキャンしてビット列を生成し、そのビット列を符号化する場合についてさまざまなスキャンの例と共に説明したが、この第7実施形態ではこれらの複数のスキャン方法のうち、最適なものを1つ選択してその出力ビット列を符号化する。

30

#### 【0076】

図19の場合と同様に、フレーム毎に整数化部12（例えば図1参照）からのそれぞれのサンプルに対応する桁数 $n(n_0, n_1, n_2, \dots, n_{1023})$ が記憶部17Bに格納され、制御部17Cの制御により差分信号Zの仮数部Mの非ゼロとなり得るビットである下位 $h = 23 - n$ ビットがバッファ17A1、17A2、17A3に取り込まれ、全てのバッファに同じように、例えば図18に示したようにMSB側に位置をそろえて格納される。制御部17Cはバッファ17A1に格納されたビット配列に対しては、図18に示したように振幅方向の各ビット位置で時間方向にスキャンして1つのビット列 $E_{j1}$ を生成する。バッファ17A2に格納されたビット配列に対しては、例えば図21に示すように、振幅方向の予め決めたビット幅毎に、その幅内のビットを時間方向に順次スキャンしてビット列 $E_{j2}$ を生成する。バッファ17A3に格納されているビット配列に対しては、例えば図22に示すように振幅方向の予め決めたビット幅毎にそのビット幅内で振幅方向にスキャンすることを全サンプルに対し順次連続して行うことによりビット列 $E_{j3}$ を生成する。

40

#### 【0077】

制御部17Cは、フレーム内で各スキャン方法による各ビット列 $E_j$ 中のビット値の偏りを例えば“1”の計数値として求め、それに基づいて各スキャン方法によるフレーム内の“1”の偏りの大きさを求め、どのスキャン方法によれば最も偏りが大きい(エントロ

50



ピーが小さい)かを判定し、最も偏りの大きいスキャン方法に対応するバッファからのビット列  $E_j$  を選択する選択信号  $SL$  を選択部 17E に与える。選択されたビット列は符号化部 17D で符号化され、符号列  $C_b$  として出力される。

なお、フレーム内のビット値 "1" の偏りに対応するエントロピーは、例えば各ビット列  $E_j$  のビット数を  $m_j$ 、そのビット列中の "1" の数を  $k_j$  とすると次式

$$\sum_j -m_j \left( \left( \frac{k_j}{m_j} \right) \log_2 \left( \frac{k_j}{m_j} \right) + \left( 1 - \left( \frac{k_j}{m_j} \right) \right) \log_2 \left( 1 - \left( \frac{k_j}{m_j} \right) \right) \right) \quad (2)$$

で表される。 $\sum_j$  は  $j$  の全てのとり得る値 (ビット列の数) についての加算を表す。ビット値 "1" の偏りが大きいほどこのエントロピーは小さくなる。このエントロピーは、フレーム内のビット配列をどのようにスキャンし、どのようにビット列に分割するかによって異なる。従って、この第7実施形態では、フレーム毎に非ゼロとなり得るビットの配列に対し、最適なスキャン方法を選択している。

#### 【0078】

図27の圧縮部17に対応する図7、10、13における伸張部23の構成例を図28に示す。この構成は、図20で示した伸張部23の構成において、復号部23Eが追加された構成となっている。補助符号  $C_e$  を復号部23Eにより復号して選択信号  $SL$  が得られ、制御部23Cに与えられる。制御部23Cは与えられた制御信号に対応するスキャン方法と桁数  $n$  に基づいて復号部23Dからのビット列  $E_j$  をバッファ23A内に配列格納してサンプルごとの非ゼロとなり得るビット列とする。その後、サンプルごとに非ゼロとなり得る  $h_j$  ビットの上位側に  $n_j$  ビットの "0" を付加して仮数部  $M$  とし、その仮数部  $M$  を差分信号  $Z$  として出力する。

#### 【0079】

図29は図27の変形例であり、ここではバッファ17A1, 17A2, 17A3からのビット列  $E_{j1}$ ,  $E_{j2}$ ,  $E_{j3}$  をそれぞれ符号化部17D1, 17D2, 17D3で符号化する。分析部17Gはそれらの符号化出力のうち、最も情報量の少ないものを判定し、その符号出力を選択する選択信号  $SL$  を符号化部17Fに与えて符号  $C_e$  を得ると共に、選択部17Eに与える。選択部17Eは、選択信号により指定された最も情報量の少ない符号列を符号列  $C_b$  として出力すると共に符号  $C_e$  も出力する。図29の圧縮部17に対応する復号化装置の伸張部23としては図28の伸張部23をそのまま使用することができる。

#### 【0080】

図11における圧縮部17と131の代わりに、図30に示す構成としてもよい。この構成は可変ビット圧縮部17-1と、可変ビット抽出部17-2と、仮数部圧縮部131-1と、仮数部抽出部131-2とが設けられている。可変ビット圧縮部17-1は差分信号  $Z$  の仮数部  $M$  から桁数  $n$  を使って  $h = 23 - n$  ビットの非ゼロとなり得るビット (可変ビット) を抽出し、圧縮符号化して符号  $C_b$  を出力する。可変ビット抽出部17-2は差分信号  $Z$  の仮数部  $M$  から  $h$  ビットの非ゼロとなり得るビット列  $M_D$  を抽出し、そのまま出力する。仮数部抽出部131-1は差分信号  $Z$  の仮数部  $M$  を圧縮符号化し、符号  $C_m$  を出力する。仮数部抽出部131-2は差分信号  $Z$  の仮数部  $M$  を抽出し、そのまま出力する。効率推定部134は、これら  $C_b$ ,  $M_D$ ,  $C_m$ ,  $M$  及び  $Z$  の情報量をそれぞれ計算し、最も情報量の少ないものを判定し、それを選択する選択信号  $SL$  を補助符号化部103で符号化すると共に出力部104に与える。出力部104は与えられた  $C_b$ ,  $M_D$ ,  $C_m$ ,  $M$ ,  $Z$  のうち、選択信号  $SL$  に対応するものを符号  $C_b$  として選択出力すると共に符号  $C_e$  も出力する。

#### 【0081】

図31は図30の構成に対応して、図10の復号化装置における伸張部23、231、選択部233、補助復号部234に置き換えられる。ただし、分析部232は不要である。図31において、符号  $C_e$  は復号部234で復号されて選択信号  $SL$  が得られ、これにより選択部233を制御する。即ち、入力符号  $C_b$  を可変ビット列の圧縮符号  $C_b$  として復号部23に与えるか、そのまま可変ビット列 (非ゼロとなり得るビット列)  $M_D$  として

組立部 2 3 a に与えるか、仮数部 M の圧縮符号  $C_m$  として復号部 2 3 1 に与えるか、仮数部 M そのものとして出力するか、のいずれか S L に対応する処理を実行する。復号部 2 3 は符号列  $C_b$  を復号して非ゼロとなり得るビット列  $M_o$  を生成し、組立部 2 3 a に与える。組立部 2 3 a は、桁数  $n$  に基づいて  $n$  ビットの " 0 " を非ゼロとなり得るビット列  $M_o$  の上位に付加して仮数部 M を生成する。

#### 変形例

図 5 に示した関係は切り捨て、つまり必ず絶対値が小さくなるように入力浮動小数点信号サンプル系列 X の各サンプルを整数値としたが、浮動小数点化部 1 5 で浮動小数点信号サンプル系列 Y に再変換したあとでも指数部が、入力浮動小数点信号サンプル系列 X の対応サンプル間で変わらないようにすれば、必ずしも切り捨てで整数にする必要はなく四捨五入でもよい。図 3 2 に示すように、 $0 \sim 1 \sim 2 \sim \dots \sim 9$  と連続した浮動小数点入力に対し、切り捨てにより整数値にすると、 $0, 1, 2, \dots, 9$  となる。従って入力浮動小数点から整数値を引いた誤差信号、つまり差分信号 Z の仮数部は必ず正になる。これに対し、図 3 3 に示すように四捨五入では  $0 \sim 0.5 \sim 1.5 \sim 2.5 \sim \dots \sim 9.5$  と連続した入力浮動小数点が入力され、 $0, 1, 2, \dots, 9$  となる。従ってもとの浮動小数点から再変換したあとの浮動小数点の値を引いた誤差信号、つまり差分信号 Z は負の数にもなりうる。整数化の際に、2 のべき乗の値に切り上げが生じた場合は、再変換した浮動小数点信号サンプル系列 Y の各サンプルの指数部が、入力浮動小数点デジタル信号系列 X のサンプルに対し変化し、仮数部は入力浮動小数点信号サンプル系列 X のそれとまったく異なるので圧縮には不利である。

#### 【0082】

そこで以下に示す変形四捨五入を考える。図 3 4 に示すように、 $0 \sim 1 \sim 2 \sim 2.5 \sim 4 \sim 4.5 \sim 5.5 \sim 6.5 \sim 8 \sim 9.5 \sim \dots$  と連続した入力浮動小数点を  $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \dots$  と整数化し、つまり図中に矢印で示すように、仮数部 M の  $n$  ビットの整数部分  $M_e$  が 2 のべき乗の直前の値  $2^r - 1$  のときは端数部分  $M_f$  を四捨五入ではなく切り捨てを行い、その他の場合は四捨五入する。誤差信号 ( 差分信号 ) Z の正負の区別の極性に対して 1 ビットが必要になるが、四捨五入したサンプルについては誤差が  $0.5$  以下になるので仮数部の最大桁数は 1 ビット減る。このため極性もあわせた仮数部の桁数は切り捨ての際と同じである。ただし整数部分  $M_e$  が 2 のべき乗値  $2^r$  の際は誤差の仮数部が一桁少なくすみ、整数部分が  $2^r - 1$  の際は誤差が正の場合のみ最大 1 までの誤差が生じるので特別扱いする必要がある。

#### 【0083】

つまり差分信号 Z の各サンプルごとの非ゼロになり得る桁の先頭ビットに極性ビットを付け、整数部分  $M_e$  が  $2^r - 1$  の場合は非ゼロになり得る桁数  $h$  を示す符号も付けて圧縮符号化を行い、復号装置において、符号列  $C_b$  を復号伸張したビット列から、整数部分の桁数  $n$  から求めた非ゼロになり得るビット数  $h$  と極性の 1 ビットを加えたビット数毎に分離し、整数部分が  $2^r - 1$  の場合は、極性ビットと、非ゼロになり得る桁数  $h$  を示す符号と、その符号が示すビット数  $h$  との合計ビット数を分離して差分信号 Z の 1 サンプルを求める。なお、非ゼロとなり得る桁のみを符号化する場合は、振幅が大きい場合に効率がよく、振幅が大きい場合、 $2^r - 1$  となる回数、つまり前記例外処理を行う回数は小さくなり、全体として圧縮効率を上げることができる。

#### 【0084】

以上のように変形四捨五入をすることにより、浮動小数点入力と整数部分の誤差のエネルギーは切り捨てより小さくでき、整数部分の系列の予測効率が改善される場合があり、しかも圧縮部 1 7 において差分信号 Z を、非ゼロになり得る桁についてのみ符号化することにより圧縮効率を高くすることができる。

この変形四捨五入による整数化は例えば、図 3 中に 1 点鎖線で示すように、ステップ S 2 の次に、入力浮動小数点信号サンプル系列 X の整数部分  $M_e$  は  $2^r - 1$  であるかを調べ、例えば仮数部 M の上位  $n = E - E_o$  ビットが全て 1 かを調べ、全て 1 であれば、2 のべき乗の直前の値であるから、ステップ S 3 に切り捨て処理を行い、上位  $E - E_o$  ビッ

トが全て1でなければステップS6に移って、四捨五入処理を行う。

**変形実施形態**

図9に示した第2実施形態において、分析部132の判定結果に応じて圧縮部内による圧縮符号化か、圧縮部131による圧縮符号化を選択した。この選択を正確に行うには図11に示したように実際に両圧縮部17と131でそれぞれ符号化し、その結果により、圧縮効率が高い方の圧縮符号化の結果を採用する。後者の場合は正確であるが差分信号Zを直接圧縮符号化するには比較的大きな処理量と時間を必要とする。従って以下のようにしてもよい。

**【0085】**

バイト(8ビット)単位でのユニバーサル符号化の効率を推定する方法の一例として、バイトごとのエントロピーの計算をする方法がある。誤差信号つまり浮動小数点形式の差分信号Zを8ビット単位で0から $2^8 - 1 (= 255)$ の整数yとみなし、その1フレーム内におけるyの出現回数L(y)からヒストグラムを作り、フレーム内のバイト数Bで割り算をしてyについての密度関数p(y)を求める。

10

$$p(y) = L(y) / B \tag{3}$$

この密度関数p(y)から平均エントロピーH(p(y))を求める。

**【0086】**

[数2]

$$H(p(y)) = - \sum_{y=0}^{255} p(y) \log_2 p(y) \tag{4}$$

20

**【0087】**

ユニバーサル符号化では1バイトの数値をこの平均エントロピー(ビット)程度までに圧縮できる可能性があるので、この平均エントロピー値をフレーム内のバイト数B倍した値が、圧縮部17による0以外になり得る桁についてのみを圧縮符号化した場合の1フレーム内の符号化ビット数の合計値より小さい場合には、バイト単位のユニバーサル符号化、つまり圧縮部131による圧縮符号化の方が有利となる。

このような圧縮部の選択を行うことができる構成を図35に示す。図35に示す符号化装置は図11の構成に選択部133と、平均エントロピー計算部136と、判定部137とを加えたものである。図に示すように、差分生成部14からの差分信号Zを平均エントロピー計算部136に入力して、先に述べたように、平均エントロピーH(p(y))を求め、このH(p(y))をバイト数B倍した値を求め、これと効率推定部134aで推定された符号列Cb'の1フレーム分のビット数と判定部137で比較し、前者の方が小さければ、選択部133を制御して差分信号Zを圧縮部131で圧縮符号化し、後者の方が小さければ、圧縮部17で圧縮符号化した符号列Cb'を符号列Cbとして出力する。またいずれの圧縮部を選択したかを示す補助符号Ceも出力する。なお前記エントロピーの計算の際に整数化部12において切り捨てにより整数化を行った場合は、差分信号Zの各サンプルの仮数部Mの23ビットを1バイト単位で出現回数L(y)を求め、変形四捨五入の場合は各サンプルについて極性を含めて出現回数L(y)を求める。

30

40

**【0088】**

以上のように、ユニバーサル符号化の効率を推定すれば、差分信号Zをユニバーサル符号化して、その1フレーム内での全符号化ビット数を求めて、分析選択部134で圧縮部を選択する場合よりも、処理量を少なくすることができる。このようなユニバーサル符号化の効率推定の利用は図8に示した実施形態にも適用することもできる。

上述において、整数形式信号サンプル系列Yを1サンプル16ビット、24ビットの場合としたが、これらは24ビット以下の任意の数としてもよく、その数に応じて整数形式デジタル信号系列Yのサンプル当りのビット数変更に基づく、ゼロ以外になり得る桁数を決定すればよい。このように分割単位毎に圧縮符号化する場合は、整数化するビット数

50

を分割単位毎に都合のよいように決め、そのビット数を指定する補助符号も出力するようにしてもよい。

#### コンピュータによる実施形態

図1、8、9、11、12、16、35に示した符号化装置は図36に示すように、バス68で互いに接続されたRAM61、CPU62、ハードディスク63、ROM64、送受信部65、入出力部66、オーディオ・ビデオ入出力部67などを有するコンピュータ600により実施させてもよい。

##### 【0089】

ROM64にはコンピュータの起動に関するプログラムが格納されており、ハードディスク63にはコンピュータの動作システムプログラムが格納されている。この発明の符号化装置の機能構成を実行するプログラムは予めCD-ROM又はDVDなどの記録媒体71からCD/DVDドライブ66によりハードディスク63に読み込んでおくか、通信回線を通して送受信部65によりダウンロードしてハードディスク63に読み込んでおく。

符号化する浮動小数点形式の信号サンプル系列Xは例えばオーディオ・ビデオ入出力部67により外部から取り込んで、バッファとして動作させるハードディスク63に一時記憶する。符号化を開始するには、符号化を実行するプログラムをハードディスク63からRAM61に読み込んで、その実行プログラムをCPU62により実行する。符号化結果である符号Ca、Cb(又はZ)、Cc等は送受信部65により通信回線を通して送信してもよいし、CD/DVDドライブ66によりCD又はDVDの記録媒体に記録保存してもよい。

##### 【0090】

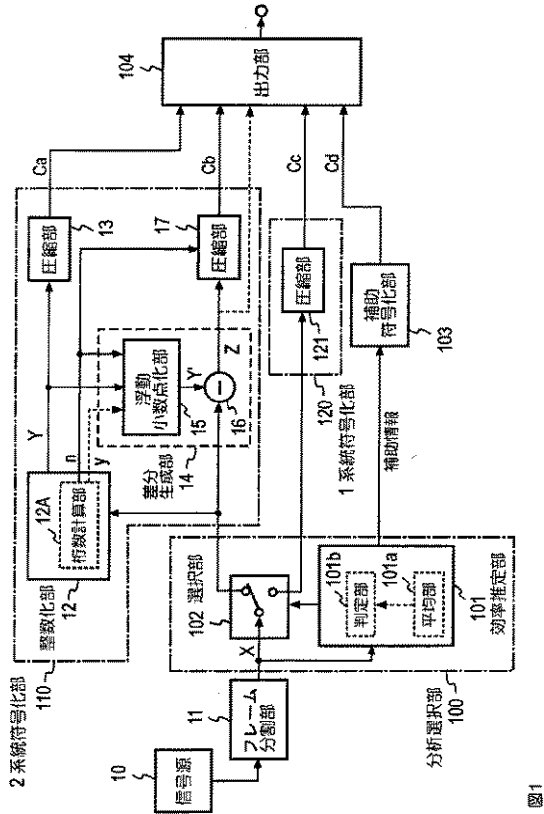
図7、10、11、13、17に示した復号化装置を図36のコンピュータで実施する場合も同様に、復号化装置の機能構成を実行するプログラムをハードディスク63に格納しておき、送受信部65で通信回線から受信した符号Ca、Cb(又はZ)、Ccに対し復号プログラムを実行する。復号結果は表示器73及びスピーカ74に再生出力する。

この発明は音楽信号のみならず、音声信号、画像信号などに対しても適用することができる。

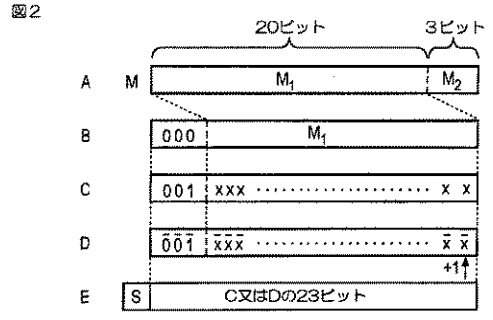
10

20

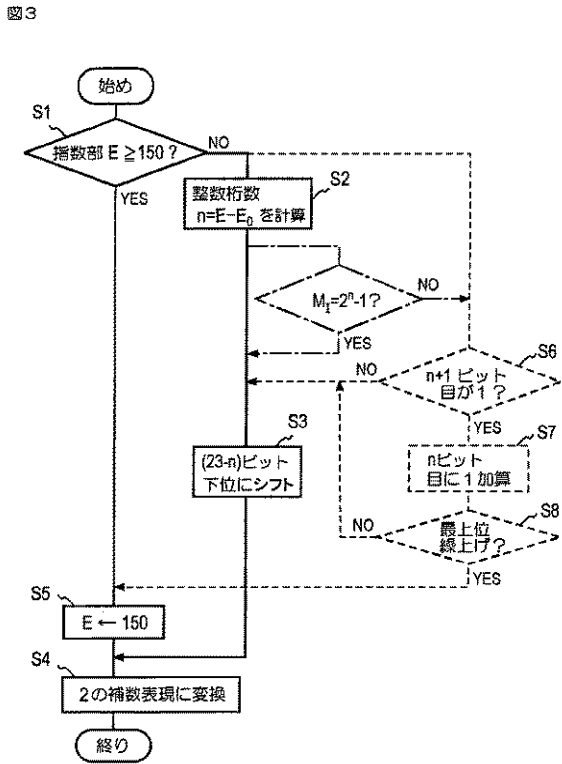
【図1】



【図2】



【図3】



【図4】

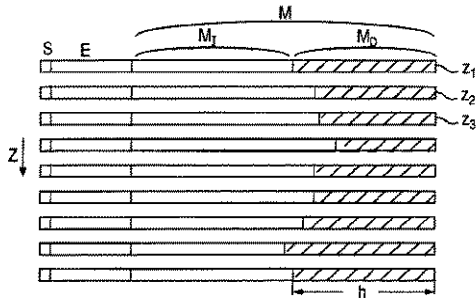
n	整数値の絶対値 $X_W$	仮数部 M 23ビット
-127	0	$0 \cdot x_{23} \ x_{22} \ \dots \ x_1$
0	1	$1 \cdot x_{23} \ x_{22} \ \dots \ x_1$
1	2~3	$1 \cdot x_{23} \cdot x_{22} \ \dots \ x_1$ 2ビット
2	4~7	$1 \cdot x_{23} \cdot x_{22} \cdot x_{21} \ \dots \ x_1$ 3ビット
k-1	$2^{k-1} \sim (2^k-1)$	$1 \cdot x_{23} \cdot x_{22} \ \dots \ x_{23-(k-2)} \cdot x_{23-(k-1)} \ \dots \ x_1$ kビット
22	$2^{22} \sim (2^{23}-1)$	$1 \cdot x_{23} \ \dots \ x_2 \cdot x_1$ 23ビット

【図5】

整数値の絶対値 $X_W$	極性 S と指数部 E	仮数部 M
0	制約なし	制約なし
1	0	制約なし
2~3	0	上位1桁0
4~7	0	上位2桁0
-	-	-
$2^{k-1} \sim (2^k-1)$	0	上位k-1桁0
$2^{22} \sim (2^{23}-1)$	0	上位22桁0

【図6】

図6



【図7】

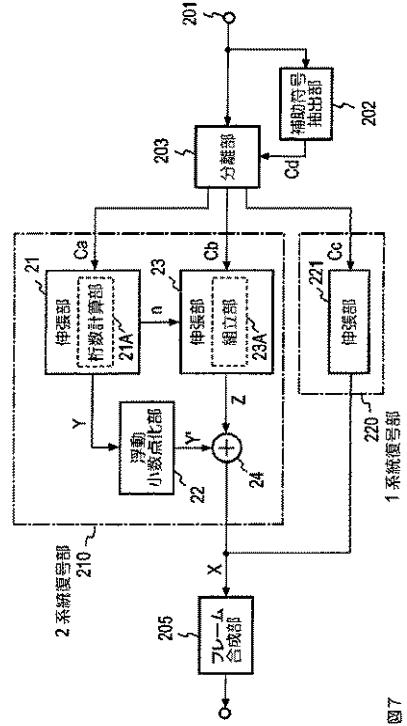


図7

【図8】

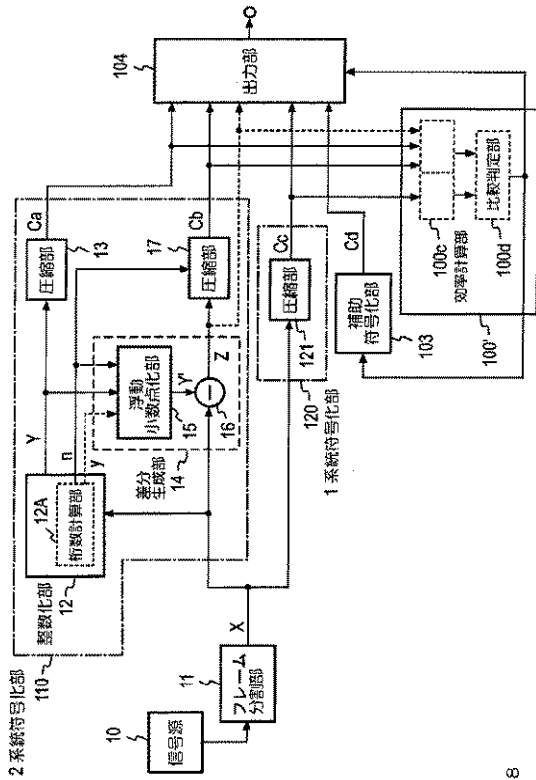


図8

【図9】

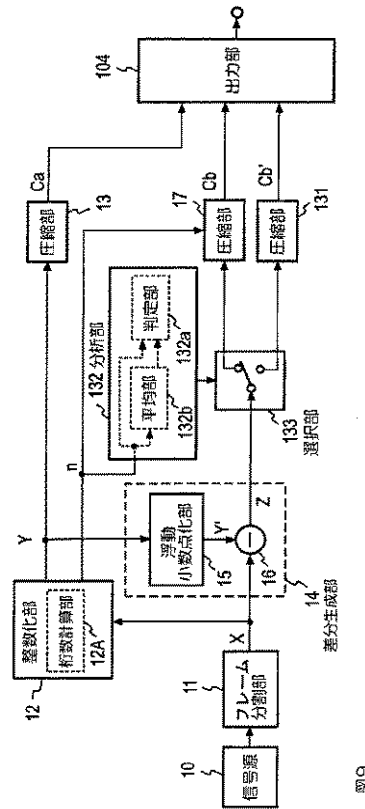


図9







【図18】

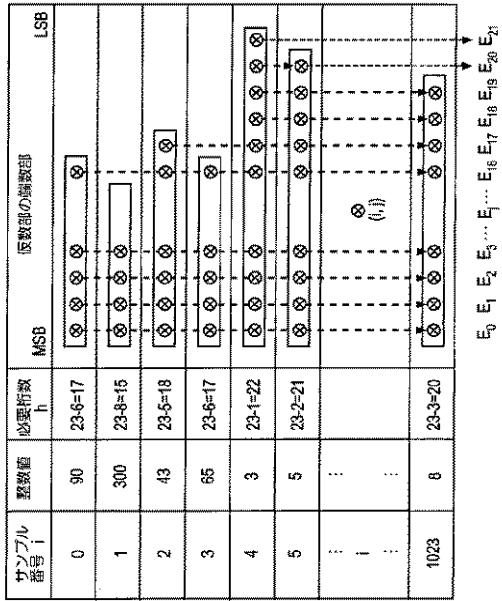
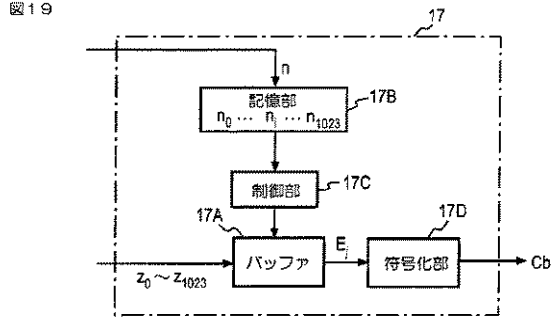


図18

【図19】



【図20】

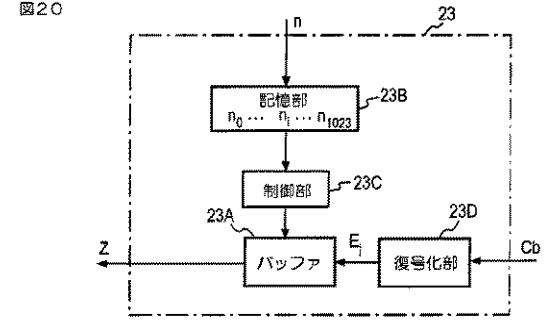


図20

【図21】

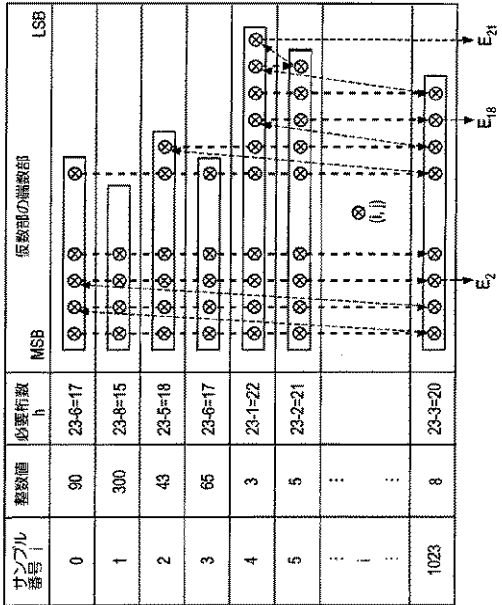


図21

【図22】

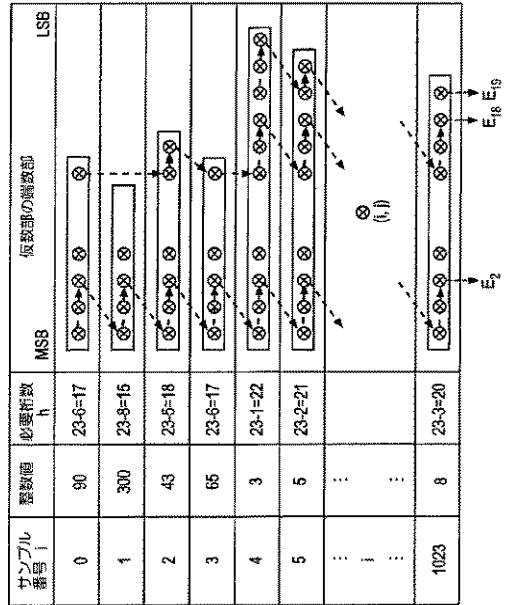


図22

【図 2 3】

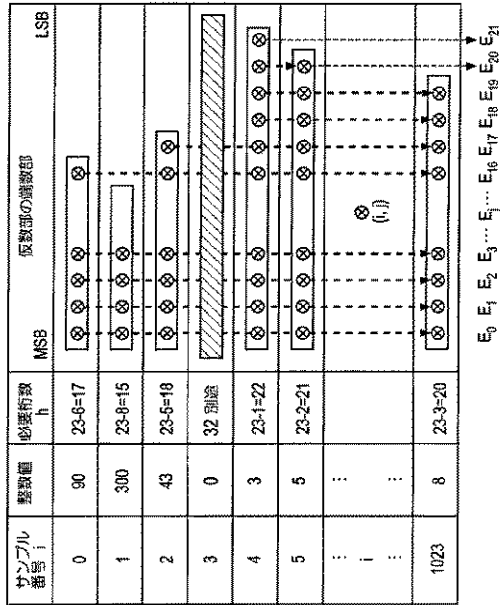


図23

【図 2 4】

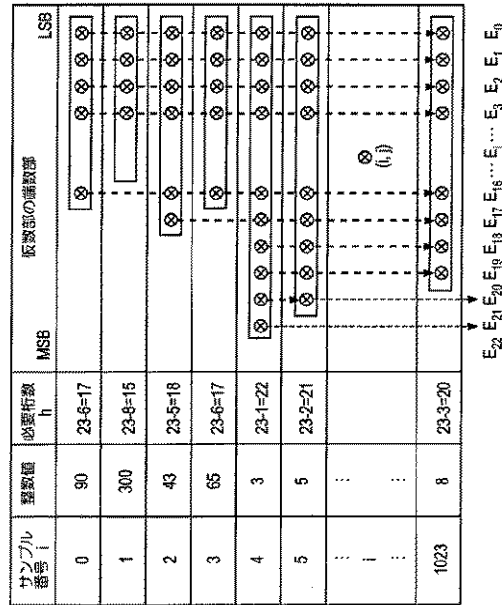
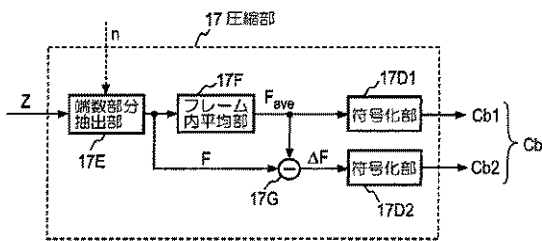


図24

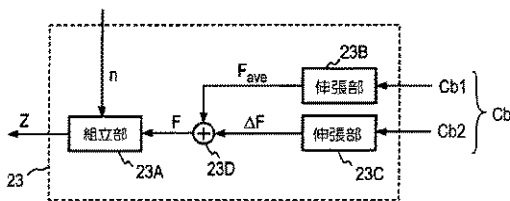
【図 2 5】

図 25



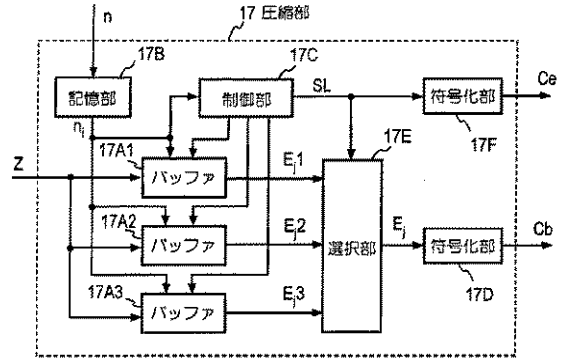
【図 2 6】

図 26



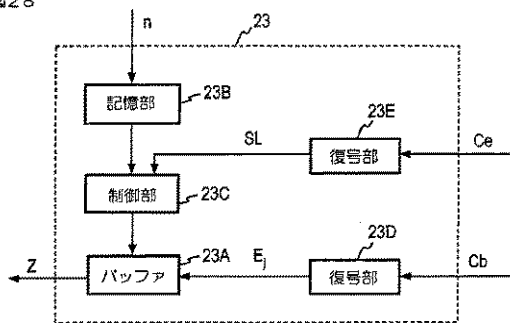
【図 2 7】

図 27



【図 2 8】

図 28



【図29】

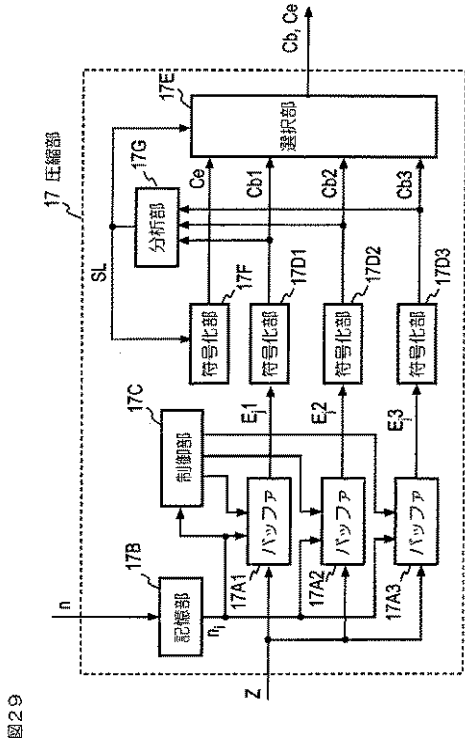


図29

【図30】

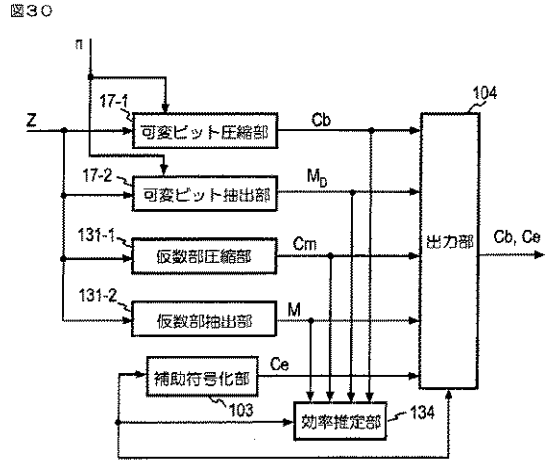


図30

【図31】

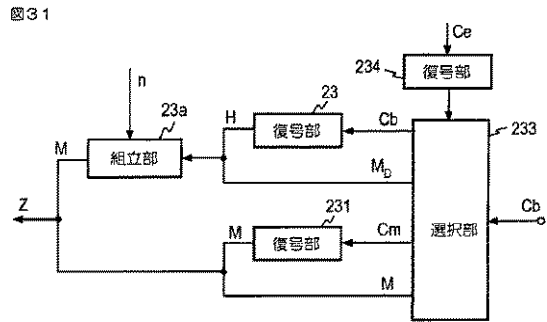


図31

【図32】

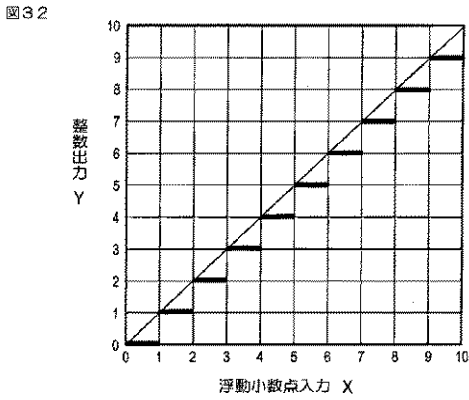


図32

【図34】

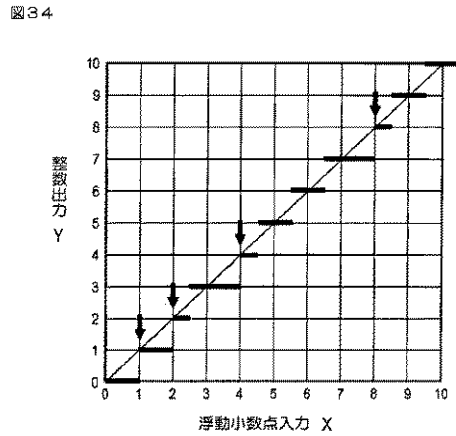


図34

【図33】

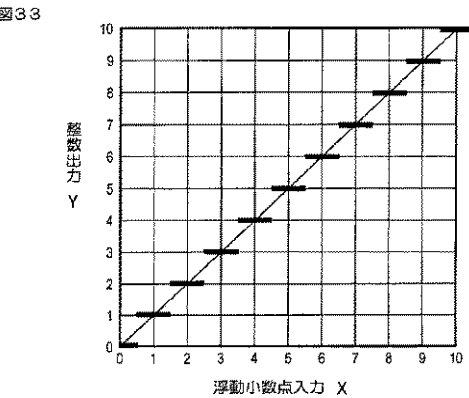


図33

【図35】

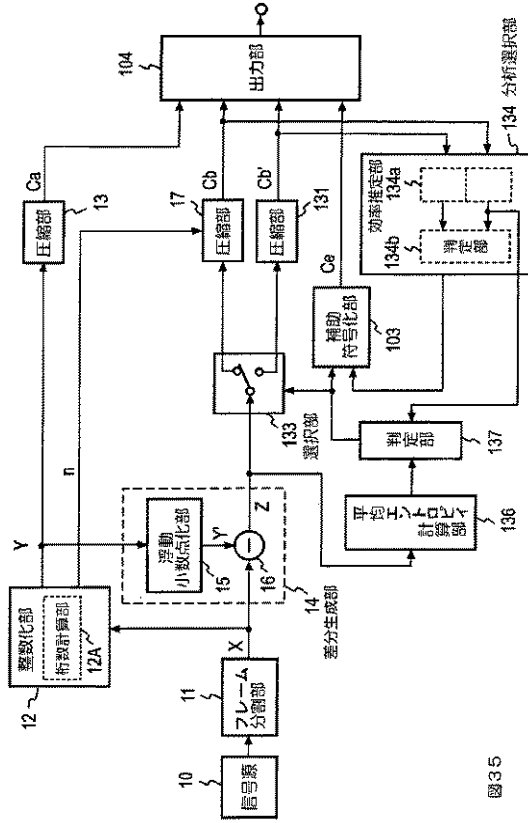
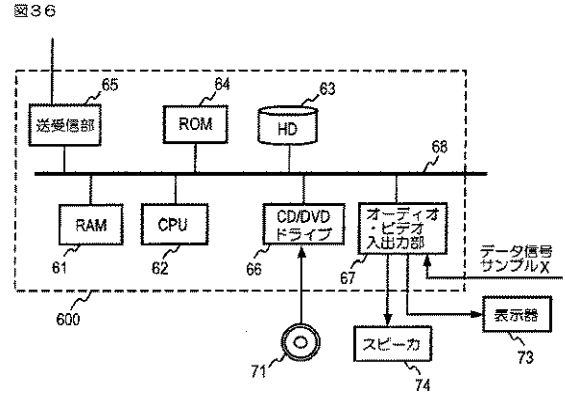


図35

【図36】



【図37】

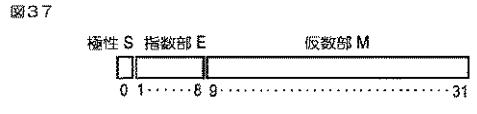


図37

-----  
フロントページの続き

(72)発明者 原田 登

東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内

審査官 渡辺 未央子

(56)参考文献 特開2001-044847(JP,A)

特開2003-332914(JP,A)

国際公開第03/077425(WO,A2)

特開平07-199996(JP,A)

特開平04-180413(JP,A)

守谷健弘, 他, 音響信号のサンプリングレート・スケーラブル・ロスレス符号化, 情報科学技術  
フォーラム一般講演論文集, 情報処理学会, 2002年 9月13日, 第2分冊, pp.227-228

Akio Jin et al., A Hierarchical Lossless/Lossy Coding System for High Quality Audio up  
to 192 kHz Sampling 24 Bit Format, IEEE Transactions on Consumer Electronics, IEEE,  
2003年 8月, Vol.49, No.3, pp.759-764

(58)調査した分野(Int.Cl., DB名)

H03M 7/36

H03M 7/24

G10L 19/00

H04N 7/26