

(19) 日本国特許庁 (JP)

(12) 特許公報 (B2)

(11) 特許番号

特許第4049792号

(P4049792)

(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.⁸

F I

H 0 3 M 7/36 (2006.01)

H 0 3 M 7/36

請求項の数 2 8

(全 2 8 頁)

(21) 出願番号 特願2005-507254 (P2005-507254)
 (86) (22) 出願日 平成16年6月21日 (2004.6.21)
 (86) 国際出願番号 PCT/JP2004/008726
 (87) 国際公開番号 WO2004/114527
 (87) 国際公開日 平成16年12月29日 (2004.12.29)
 審査請求日 平成17年8月5日 (2005.8.5)
 (31) 優先権主張番号 特願2003-176224 (P2003-176224)
 (32) 優先日 平成15年6月20日 (2003.6.20)
 (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】 浮動小数点形式デジタル信号可逆符号化方法、及び復号化方法と、その各装置、その各プログラム

(57) 【特許請求の範囲】

【請求項 1】

(a) 浮動小数点形式の第 1 信号サンプルを絶対値が小さくなるように切捨てを行って整数形式の第 2 信号サンプルに変換する過程と、

(b) 上記整数形式の第 2 信号サンプルを可逆圧縮して第 1 符号列を生成して出力する過程と、

(c) 上記整数形式の第 2 信号サンプルと上記浮動小数点形式の第 1 信号サンプルとの差分に対応する浮動小数点形式の差分信号を生成する過程と、

(d) 上記浮動小数点形式の差分信号の、上記整数形式の第 2 信号サンプルの絶対値により決まる非ゼロになり得るビット領域に対応する第 2 符号列を生成して出力する過程、
 とを含む浮動小数点形式デジタル信号可逆符号化方法。

10

【請求項 2】

請求項1記載の符号化方法において、上記過程(d) は、上記差分信号の、上記非ゼロになり得るビット領域を可逆圧縮して上記第 2 符号列を生成する過程を含む。

【請求項 3】

請求項 1 または 2 記載の符号化方法において、上記非ゼロになり得るビット領域は、上記第1信号サンプルの仮数部における小数点以下の端数部分である。

【請求項 4】

請求項 1 または 2 記載の符号化方法において、上記過程(a) は、

(a-1) 上記第 1 信号サンプルの複数サンプル区間毎にその区間内の上記浮動小数点形式

20

の第 1 信号サンプルの指数部の最大値を検出する過程と、

(a-2) 上記区間内の上記第 1 信号サンプルの整数値に変換後の最大桁数が予め決めた値と成るように上記区間内の第 1 信号サンプルの上記指数部の値をそれぞれ上記検出した最大値に基いて決めた調整値だけ調整する過程と、

(a-3) 上記指数部が調整された浮動小数点形式の上記区間内の第 1 信号サンプルを、上記整数形式に変換する過程と、

(a-4) 上記調整値と対応した調整情報を符号化して補助符号を生成する過程、
とを含み、上記過程(d) は上記補助符号も出力する。

【請求項 5】

請求項 2 記載の符号化方法において、上記整数形式の第 2 信号サンプルのビット数 b_1 、
は上記浮動小数点形式の第 1 信号サンプルの仮数部のビット数 b_M 以下であり、

上記第 2 符号列の生成過程は、上記整数形式の第 2 信号サンプルの絶対値に対応して決まる上記第 1 信号サンプルの仮数部における非ゼロになり得る多数部分の領域を、上記仮数部のビット数 b_M と上記第 2 信号サンプルのビット数 b_1 との差のビット数の上位側ビット領域と、残りの下位側ビット領域とに分けてそれぞれ可逆圧縮し、それらの符号を上記第 2 符号列として出力する過程を含む。

【請求項 6】

請求項 2 記載の符号化方法において、上記過程(d) は、フレーム毎に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それらの最上位ビットをそろえて配置し、少なくともサンプル方向にビットを選択して複数のビット列を生成し、それぞれの上記ビット列を可逆圧縮して上記第 2 符号列を生成する過程を含む。

【請求項 7】

請求項 1 記載の符号化方法において、上記ステップ(c) は、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、サンプル方向に予測を行い、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 符号列とする過程である。

【請求項 8】

(a) 第 1 符号列を復号伸張して整数形式の第 1 信号サンプルを生成する過程と、

(b) 上記第 1 信号サンプルの絶対値に応じて決まる非ゼロになり得る桁数に基いて、第 2 符号列から浮動小数点形式の差分信号を生成する過程と、

(c) 上記整数形式の第 1 信号サンプルを浮動小数点形式の第 2 信号サンプルに変換する過程と、

(d) 上記浮動小数点形式の第 2 信号サンプル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第 3 信号サンプルを生成する過程、
とを含む浮動小数点形式デジタル信号復号化方法。

【請求項 9】

請求項 8 記載の復号化方法において、上記差分信号を生成する過程(b) は、上記第 2 符号列を復号伸張して上記非ゼロになり得る桁の端数部分を生成し、上記端数部分を使って上記浮動小数点形式の差分信号を生成する過程である。

【請求項 10】

請求項 8 または 9 記載の復号化方法において、補助符号を復号して調整情報を生成する過程と、上記整数形式の第 1 信号サンプル又は上記合成信号に対し、上記調整情報により桁調整する過程とを含む。

【請求項 11】

請求項 9 記載の復号方法において、上記第 2 符号列は第 1 符号系列と第 2 符号系列とを含み、

上記差分信号の生成過程(b) は、

(b-1) 上記整数形式の第 1 信号サンプルのビット数と上記浮動小数点形式の第 3 信号サ

ンプルの仮数部のビット数との差のビット数の上位側ビット領域に上記第 1 符号系列を復号伸張する過程と、

(b-2) 上記第 2 符号系列を下位側ビット領域に復号伸張する過程と、

(b-3) 上記上位側ビット領域の下位側に上記下位側ビット領域を付加して仮数部の端数部分として上記差分信号を生成する過程、
とを含む。

【請求項 1 2】

請求項 9 記載の復号化方法において、上記過程(b) は、

(b-1) フレーム毎に上記第 2 符号列を復号伸張して複数のビット列を得る過程と、

(b-2) フレーム内のサンプル方向におけるサンプル位置とサンプル内の振幅方向における位置で規定されるビット位置を少なくともサンプル方向に走査して各ビット位置が対応する上記整数形式の第 1 信号サンプルの絶対値により決まる端数部分のビット長内にあるビット位置に各上記ビット列のビットを分配してフレーム内の差分信号の端数部分を再構成する過程と、

(b-3) 各上記端数部分の上位側に上記対応する第 1 信号サンプルの上記絶対値により決まる仮数部のビット長内の端数部分以外のビット数だけゼロを挿入して各差分信号の仮数部を再構成することにより、上記差分信号を生成する過程、
とを含む。

【請求項 1 3】

請求項 8 記載の復号化方法において、上記過程(b) は、上記第 2 符号列を伸張復号してフレームごとのサンプルの平均値とフレーム内の各サンプルに対応する予測誤差から平均値が除かれた信号を得る過程と、上記平均値と平均値が除かれた信号を加算して端数部分を生成する過程と、上記端数部分の上位側に、上記整数形式の第 1 信号サンプルの絶対値により決まる仮数部のビット長内の端数部分以外のビット数だけ 0 を挿入して差分信号の仮数部を生成する過程とを含む。

【請求項 1 4】

入力された浮動小数点形式の第 1 信号サンプルを絶対値が小さくなるように切捨てを行って整数形式の第 2 信号サンプルに変換する整数化部と、

上記整数形式の第 2 信号サンプルが入力され、これを可逆圧縮して第 1 符号列を生成する第 1 圧縮部と、

上記整数形式の第 2 信号サンプルと上記浮動小数点形式の第 1 信号サンプルとの差分に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、

上記浮動小数点形式の差分信号の、上記整数形式の第 2 信号サンプルの絶対値に応じて決まる非ゼロになり得るビット領域に対応する第 2 符号列を生成して出力する第 2 符号列生成手段、

とを含む浮動小数点形式デジタル信号可逆符号化装置。

【請求項 1 5】

請求項 1 4 に記載の符号化装置において、上記第 2 符号列生成手段は、上記差分信号を可逆圧縮し、圧縮結果を上記第 2 符号列として出力する第 2 圧縮部を含む。

【請求項 1 6】

請求項 1 4 または 1 5 記載の符号化装置において、

上記第 1 デジタル信号サンプルの複数サンプル区間毎にその区間内の上記浮動小数点形式の第 1 デジタル信号サンプルの指数部の最大値を検出し、上記区間内の上記第 1 デジタル信号サンプルの整数値に変換後の最大桁数が予め決めた値となるように上記区間内の第 1 デジタル信号サンプルの上記指数部の値をそれぞれ上記検出した最大値に基づいて決めた調整値だけ調整する桁調整部と、

上記調整値と対応した調整情報を符号化して補助符号を生成出力する補助符号化部、
とを含む。

【請求項 1 7】

請求項 1 5 記載の符号化装置において、上記整数形式の第 2 信号サンプルのビット数 b

は上記浮動小数点形式の第 1 信号サンプルの仮数部のビット数 b_M 以下であり、

上記第 2 圧縮部は、上記整数形式の第 2 信号サンプルの絶対値に対応して決まる上記第 1 信号サンプルの仮数部における非ゼロになり得る端数部分の領域中の、上記仮数部のビット数 b_M と上記第 2 信号サンプルのビット数 b_1 との差のビット数の上位側ビット領域を可逆圧縮符号化して第 1 符号系列を生成する第 1 圧縮符号化部と、上記端数部分の上記上位側ビット領域の下位側のビット領域を可逆圧縮符号化して第 2 符号系列を生成する第 2 圧縮符号化部とを含み、それら第 1 及び第 2 符号系列は上記第 2 符号列として出力される。

【請求項 18】

請求項 15 記載の符号化装置において、上記差分生成部は、フレーム毎に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それらの最上位ビットをそろえて配置し、少なくともサンプル方向にビットを選択して複数のビット列を生成するビット列生成主段を含み、上記第 2 圧縮部は上記ビット列を可逆圧縮して上記第 2 符号列を生成する。

10

【請求項 19】

請求項 15 記載の符号化装置において、上記差分生成部は、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る平均値算出部と、それぞれの整数値サンプルと上記平均値との誤差を求める減算器と、上記誤差をサンプル方向に予測を行い、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 符号列とする符号化部とを含む。

20

【請求項 20】

第 1 符号列が入力され、これを復号伸張して整数形式の第 1 信号サンプルを生成する第 1 伸張部と、

上記第 1 信号サンプルの絶対値に応じて決まる非ゼロになり得る桁数に基づいて、第 2 符号列から浮動小数点形式の差分信号を生成する差分信号復号手段と、

上記整数形式の第 1 信号サンプルが入力され、これを浮動小数点形式の第 2 信号サンプルに変換する浮動小数点化部と、

上記浮動小数点形式の第 2 信号サンプルと上記浮動小数点形式の差分信号が入力され、これらを合成して浮動小数点形式の第 3 信号サンプルを生成する合成部と

を有する浮動小数点形式デジタル信号復号化装置。

30

【請求項 21】

請求項 20 記載の復号化装置において、上記差分信号復号手段は、上記第 2 符号列を復号伸張して上記非ゼロになり得る桁の端数部分を生成し、上記端数部分を使って上記浮動小数点形式の差分信号を生成する手段である。

【請求項 22】

請求項 20 または 21 記載の復号化装置において、補助符号を復号して調整情報を生成する補助復号部と、上記整数形式の第 1 信号サンプル又は上記合成信号に対し、上記調整情報により桁調整する桁調整部とを含む。

【請求項 23】

請求項 20 記載の復号化装置において、上記第 2 符号列は第 1 符号系列と第 2 符号系列とを含み、

40

上記差分信号復号手段は、上記整数形式の第 1 信号サンプルのビット数と上記浮動小数点形式の第 3 信号サンプルの仮数部のビット数との差のビット数の上位側ビット領域に上記第 1 符号系列を復号伸張する第 2 伸張部と、上記第 2 符号系列を下位側ビット領域に復号伸張する第 3 伸張部と、上記上位側ビット領域の下位側に上記下位側ビット領域を付加して仮数部の端数部分として上記差分信号を生成する合成部、とを含む。

【請求項 24】

請求項 20 記載の復号化装置において、上記差分信号復号手段は、フレーム毎に上記第 2 符号列を復号伸張復号して複数のビット列を得る復号化部と、フレーム内のサンプル方向におけるサンプル位置とサンプル内の振幅方向における位置で規定されるビット位置を

50

少なくともサンプル方向に走査して各ビット位置が対応する上記整数形式の第1信号サンプルの絶対値により決まる端数部分のビット長内にあるビット位置に各上記ビット列のビットを分配してフレーム内の差分信号の端数部分を再構成し、各上記端数部分の上位側に上記対応するサンプルの絶対値により決まる仮数部のビット長内の端数部分以外のビット数だけゼロを挿入して各差分信号の仮数部を再構成し、上記差分信号を生成する再構成手段とを含む。

【請求項 25】

請求項 20 記載の復号化装置において、上記差分信号復号手段は、上記第 2 符号列を伸張復号してフレームごとのサンプルの平均値とフレーム内の各サンプルに対応する誤差を得る伸張復号部と、上記平均値と各誤差を加算して端数部分を生成する加算部と、上記端数部分の上位側に、上記第 1 信号サンプルの絶対値により決まる仮数部のビット長内の端数部分以外のビット数だけ 0 を挿入して差分信号の仮数部を生成する手段とを含む。

10

【請求項 26】

請求項 1 乃至 7 のいずれかに記載した浮動小数点形式デジタル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 27】

請求項 8 乃至 13 のいずれかに記載した浮動小数点形式デジタル信号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 28】

請求項 26 又は 27 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。

20

【発明の詳細な説明】

【技術分野】

【0001】

この発明は音声、音楽、画像などのデジタル信号を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。

30

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。

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、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、このフレーム単位ごと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。例えば、上記デジタル入力信号が音声であれば ITU-T 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 Twin VQ (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

【0003】

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

50

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、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は全て0のビット列となるので、予め決めた短い符号で表すことができ、誤差信号の可逆圧縮符号化効率を上げることができる。

10

【0004】

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表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。

【0005】

20

音声、画像などの情報の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1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

【0006】

30

[数1]

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

【0007】

IEEE-754によれば、 $E_0 = 2^7 - 1 = 127$ 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 $a1110$ 、 $E - E_0 = 128$ の場合は $a1111$ と定義されている。

40

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、前述の非特許文献1に示されている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。

【特許文献1】日本国特許出願公開2001-44847号

【非特許文献1】Mat Hans, Ronald W. Schaffer, Lossless Compression of Digital Audio, IEEE SIGNAL PROCESSING MAGAZINE, July 2001, p. 21

50

- 3 2

【発明の開示】

【発明が解決しようとする課題】

【0008】

この発明の目的は浮動小数点形式のデジタル信号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を提供することにある。

この発明の符号化方法及び装置によれば、浮動小数点形式の第1信号サンプルを絶対値が小さくなるように切捨てを行って整数形式の第2信号サンプルに変換し、上記整数形式の第2信号サンプルを可逆圧縮して第1符号列を生成して出力し、上記整数形式の第2信号サンプルと上記浮動小数点形式の第1信号サンプルとの差分に対応する浮動小数点形式の差分信号を生成し、上記浮動小数点形式の差分信号の、上記整数形式の第2信号サンプルの最上位の1より下位のビット数により決まる非ゼロになり得るビット領域に対応する第2符号列を生成して出力する。

【0009】

このように整数形式の第1信号サンプルに変換することにより、原アナログ波形信号に近いものとなり、信号サンプル間の相関による冗長性を除く圧縮方法により効率よく圧縮することができ、差分信号は非ゼロになり得る桁についてのみ差分情報として出力するので効率が良い。

この発明の復号化方法及び装置によれば、第1符号列を復号伸張して整数形式の第1信号サンプルを生成し、上記第1信号サンプルの最上位の1より下位のビット数に応じて決まる非ゼロになり得る桁数に基づいて、第2符号列から浮動小数点形式の差分信号を生成し、上記整数形式の第1信号サンプルを浮動小数点形式の第2信号サンプルに変換し、上記浮動小数点形式の第2信号サンプル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第3信号サンプルを生成する。

【発明の効果】

【0010】

この発明によれば、浮動小数点形式のデジタル信号系列を能率よく圧縮することが可能となる。また通常の整数形式の系列の圧縮符号化モジュールを利用することができ、整数形式と浮動小数点形式の両方に対応しても処理の装置規模、プログラム規模はあまり大きくならない。

【図面の簡単な説明】

【0011】

[図1] IEEE-754の32ビット浮動小数点のフォーマットを示す図。

[図2] 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3] 図2中の整数化部12の動作を説明するための図。

[図4] 図2中の整数化部12における処理手順の例を示す図。

[図5] 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6] 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7] 差分デジタル信号系列中の圧縮符号化される部分の例を示す図。

[図8] 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] 桁調整処理手順を示すフローチャート。

[図11] 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12] 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13] この発明の第2実施形態の復号化装置の変形機能構成を示す図。

[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。

10

20

30

40

50

[図 1 4 B] 2 4 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 1 6 ビット整数化を説明するための図。

[図 1 4 C] 1 6 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 1 6 ビット整数化を説明するための図。

[図 1 5 A] 図 9 A の場合の仮数部の非ゼロとなり得る領域を示す図。

[図 1 5 B] 図 9 B の場合の仮数部の非ゼロとなり得る領域を示す図。

[図 1 6] 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 1 7] 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 1 8] 仮数部をフレーム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 1 9] 図 1 8 の符号化を実行する不圧縮部 1 7 の構成例を示す図。

[図 2 0] 図 1 9 の符号化に対応した復号化装置における伸張部 2 3 の構成例を示す図。

[図 2 1] フレーム方向のビット走査の他の例を示す図。

[図 2 2] フレーム方向のビット走査の他の例を示す図。

[図 2 3] フレーム方向のビット走査の更に他の例を示す図。

[図 2 4] 第 5 実施形態の符号化装置の構成例を示す図。

[図 2 5] 第 5 実施形態の復号化装置の構成例を示す図。

[図 2 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。

【発明を実施するための最良の形態】

【 0 0 1 2 】

第 1 実施形態

図 2 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 1 0 0 は、整数化部 1 2 と、圧縮部 1 3 と、差分生成部 1 4 と、圧縮部 1 7 とから構成されている。また、整数化部 1 2 は桁数計算部 1 2 A を有しており、差分生成部 1 4 は符号小数点化部 1 5 と、減算部 1 6 とから構成されている。信号源 1 1 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（以下簡略して入力信号サンプルとも呼ぶ）X の系列として出力される。この各デジタル信号サンプル 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、小数点以下の端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。

【 0 0 1 3 】

この浮動小数点形式の入力サンプル X は整数化部 1 2 に入力され、各サンプルごとに整数形式のデジタル信号サンプル（以下簡略して整数値信号サンプルとも呼ぶ）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の端数を切捨てる場合について図 3 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。

【 0 0 1 4 】

浮動小数点による波形の表現法として整数値の $32768 (= 2^{15})$ を 1.0 と正規化して表わす場合、あるいは整数値 2^{23} 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 3 A に示すように 2 3 ビットの仮数部 M 中の上位側 2 0 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 3 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 2 3 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 $1 \cdot M$ の 1 に対応する 1 を設定することにより、切り捨てによる整数化された値が得られる（図3C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ビットシフトしてもよい。

【0015】

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、浮動小数点形式の各デジタル信号サンプルX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3Cに得られた23ビットを、極性Sが 0（正）の場合は図3Cの23ビットをそのまま用い、Sが 1（負）であれば、図3D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3E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Yが得られる。

【0016】

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 $1 \cdot M$ 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Yとする。また、後述の他の実施形態で説明するように、フレーム毎に適正な整数の範囲に収まるように、符号小数点値の指数部を E で調整することで正規化して、その調整値 E を補助情報として符号化してもよい。

【0017】

整数化部12では例えば図4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12A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（S5）。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S4でその指数部Eを150にしてステップS5で $1 \cdot M$ 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。

【0018】

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3A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3C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 $(22 - n)$ 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Yの系列は圧縮部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Ca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ごとに予測値（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符号化すればよい。つまり整数値信号サンプルYの系列は信号源11の浮動小数点形式の入力デジタル信号サンプルXの系列の原アナログ信号波形と近似したものとなっている。従ってこの入力サンプルXの系列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。

【0019】

差分生成部14では整数値信号サンプルYと信号源11よりの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デジタル信号サンプル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Y'と原浮動小数点形式の入力信号サンプルXとの差分信号Zは、原入力信号サンプルXの小数点以下の端数値と等しい。

【0020】

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、浮動小数点形式の各入力サンプルXの極性Sと指数部E、仮数部Mの下位(23-n)ビットをそれぞれ差分Zの各サンプルの極性、指数部、仮数部の下位(23-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Yの最上位の1より下位の桁数と等しい。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Xと信号サンプルY'の仮数部の差分とする。

【0021】

圧縮部17は、整数値信号サンプルYと浮動小数点形式の入力サンプル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Cbを出力する。この圧縮部17では整数化部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。浮動小数点形式の各入力サンプルXは、

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

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5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P₀の位置を便宜的に示してある。

【0022】

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、 $-126 \leq E - E_0 \leq -1$ 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Xの仮数部23ビットが $a11 \dots 0$ である。 $-126 \leq E - E_0 \leq -1$ の場合は仮数部が $a11 \dots 0$ 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Yの全ビットは $a11 \dots 0$ となり、差分信号Zは入力サンプルXと同じになるので、浮動小数点形式の信号サンプルXの全32ビットを全て符号化する。

【0023】

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 $x_{23} \dots x_1$ は端数値 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ビットで端数値 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ビットにより端数値 M_F を表わしている。

【0024】

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 $(23 - k)$ ビットで端数値 M_F を表わす。

10

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Xから、整数値信号サンプルYを32ビット浮動小数点形式にしたサンプルY を差し引いたものであるから、差分信号Zは、図6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 - E₀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。

【0025】

整数値の絶対値 X_w が1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Xの仮数部Mと同一となる。

20

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。

【0026】

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 $x_{23} \dots x_{23 - (k-2)}$ の上位 $(k - 1)$ 桁(ビット)は必ず 0 となり、残りの $x_{23 - (k-1)} \dots x_1$ の $23 - (k - 1)$ ビットはサンプル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。整数形式デジタル信号サンプルYの整数値の範囲、つまり整数値の桁数n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。

30

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 $h = (23 - n)$ 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

40

【0027】

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XのE - E₀が-127~0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 - E₀が1~22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7に示す状態の場合、n

50

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ごとに例えばハフマン符号化などのエン

トロピ符号化を行うか、まとめて例えばユニバーサル符号化を行う。つまり図7中に斜線を施した部分のみをサンプルごとにあるいはフレーム単位など適当にまとめてエントロピ符号化などの可逆圧縮符号化を行えばよい。なお極性Sは整数形式デジタル信号サンプル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、より出力されており、受信側では整数値信号サンプルY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。

【0028】

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、差分信号Zの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。

10

なお前述したように浮動小数点形式のデジタル信号サンプル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とデジタル信号サンプルX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。

【0029】

図8に図2に示した符号化装置100と対応するこの発明による復号化装置200の実施形態を示す。

20

入力された符号列Caは伸張部21で可逆伸張復号化される。この可逆伸張復号化方法は、図2中の圧縮部13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24ビットの整数形式のデジタル信号サンプルYが生成される。また伸張部21中の桁数出力部21A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仮数部における整数部分の桁数として求め、伸張部23に与える。

【0030】

伸張部23では、入力された符号列Cbを可逆伸張復号化する。この可逆伸張復号化方法は図2中の圧縮部17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23において、伸張復号されたビット列から、組立部23Aにより桁数nに基づいて、サンプルごとの32ビット浮動小数点形式の差分(誤差)信号Zが組立てられ、出力される。図2の符号化装置100における圧縮部17で差分信号Zを圧縮しないでそのまま出力する場合は、復号化装置200の伸張部23において受信した符号列Cbをそのまま差分信号Zとして使用する。

30

【0031】

組立て部23Aでは、伸張部21からの最初のサンプルに対する桁数nと、伸張部23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、23ビットの仮数部Mの下位 $23 - n = h$ ビットとしてhビットが取り出されて、図7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32ビットの浮動小数点形式の差分信号 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7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 $23 - n = h$ 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32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7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23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 $23 - n = h$ 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云っていることに注意されたい。

40

50

【0032】

伸張部21よりの24ビット整数形式のデジタル信号サンプルYは浮動小数点化部22で1サンプル32ビットの浮動小数点形式のデジタル信号サンプルY'に変換される。このデジタル信号サンプルY'は伸張部23よりの浮動小数点形式の差分信号Zとが合成部24で対応サンプルごとに加算合成され、浮動小数点形式のデジタル信号サンプルXが再生される。各再生デジタルサンプルXの極性はデジタル信号サンプルY'のそれ、つまり復号された整数形式のデジタル信号サンプルYの極性となり、サンプルXの指数部Eは $n = 0$ 以外は、信号サンプルY'の指数部となり、仮数部Mは上位 $n = 23 - h$ ビットが信号サンプルY'の仮数部の上位 n ビットとなり、下位 $23 - n = h$ ビットが差分信号Zの下位 h ビットとなる。

10

変形実施例

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0以外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表わし、この2ビット補助情報を圧縮部13の出力符号列Caの先頭に追加し、前2者の場合は符号列Cbを出力せず、後者の場合に符号列Cbを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

前述の図2の実施形態ではE=150のサンプルに対し例外処理を行う場合を示したが、先に簡単にふれたように、フレーム毎の桁調整を行ってE=150とならないようにしてもよい。ここではフレーム内で指数部Eの最大値が $150 > E = 150 - K$ となるように桁調整を行う場合の実施形態を図9を参照して説明する。

【0033】

この第2実施形態の符号化装置100の機能構成例を図9に図2と対応する部分に同一参照番号を付けて示す。この第2実施形態では浮動小数点形式のデジタル信号サンプルXの系列を、サンプル列分割部31で複数の一定サンプル数ごとに、あるいはフレーム毎に分割し、この分割単位ごとに桁調整部32で、整数形式の信号サンプルYの桁数が適当な大きさ、つまり少なくとも整数形式に変換した際に24ビット以内に納まるように指数部Eに対して以下のように桁調整を行う。

30

(a) 浮動小数点形式のデジタル信号サンプルXの振幅が大きく、各指数部Eがもともと150を超えている場合は、24ビットの整数形式のデジタル信号サンプルYに変換すると仮数部MのLSB側1又は複数ビットの情報が失われる。あるいは浮動小数点形式のデジタル信号サンプルXの振幅が大きく、各指数部Eが150に近い値の場合は整数形式のデジタル信号サンプルYに変換した際に、そのサンプルごとのデジタル値が頻繁に24ビット整数値を超えることがある。これらのことが生じないように、デジタル信号サンプルXの指数部Eを、桁調整部32において、前記分割単位ごとに、調整情報Eだけ差し引いて150以下の値とする。

40

【0034】

(b) 浮動小数点形式のデジタル信号サンプルXの振幅が小さく、変換された整数形式のデジタル信号サンプルYの24ビット中の最上位ビットの極性Sを除き、MSBから21~23ビットが常に全て0となるような場合、振幅がわずか2ビット程度で表されることになるので整数形式のデジタル信号サンプルYの系列の波形がアナログ波形に近いものとならず、整数形式のデジタル信号サンプルYに変換する効果、つまり高い圧縮率で可逆圧縮符号化することが得られなくなる。また整数形式のデジタル信号サンプルYの振幅値がLSB側2ビット程度で表される場合、信号サンプルYを浮動小数点形式のデジタル信号に変換すると、その仮数部Mは全ビット0またはMSBのみ1で

50

他は全て 0 のように、ほとんどの桁のビットが 0 となってしまう。一方、符号小数点形式の入力デジタル信号サンプルXの仮数部Mは、信号サンプルXの振幅が小さいときでも、大きな値をとり得るので、浮動小数点形式のデジタル信号サンプルXとの誤差(差分)と対応した浮動小数点形式の差分信号 Xの仮数部Mは大きな振幅をとり、つまり全てのサンプルについて0となるビットの数が少なくなり、圧縮効率を上げることができない。

【0035】

このような問題が生じないように桁調整部32では分割単位ごとに各指数部Eに調整情報 Eだけ加算して、仮数部M中の多くの情報が整数形式のデジタル信号サンプルYに含まれるようにする。この場合、整数形式に変換した際に1サンプルのビット数が24を超えないようにする。

10

桁調整部32における調整情報 E(任意の極性の整数)は分割単位で変更が可能である。桁調整部32において分割単位ごとに指数部Eの最大を調べ、整数化した時に1サンプルが24ビットを超えない範囲で仮数部Mの情報をなるべく利用できるように調整情報 Eを決めればよい。

【0036】

図10は桁調整部32がフレーム毎に実行する桁調整の処理手順を示す。ここではサンプル列分割部31により分割された N_F 個のサンプル毎に桁調整を行う場合を示す。

ステップS1で N_F 個の入力サンプルを取り込み、ステップS2で $i = 1$ とし、 E_0 としては、取り得ない十分小さい任意の値、例えば $E_0 = -150$ に初期設定する。

20

ステップS3でi番目のサンプルの指数部 E_i から150を減算して差 E_i を得る。

ステップS4で E_i が1つ前の E_{i-1} より大きいかが判定し、大きくなければステップS6に移動する。

【0037】

ステップS5で E_i が E_{i-1} より大きい場合は、 E_i を E_{max} として一時保存する。

ステップS6で $i = N_F$ となったかが判定し、なっていないならばステップS7でiを1歩進め、ステップS3に戻る。

ステップS6で $i = N_F$ となっていれば、ステップS8で E_{max} を読み出し、1以上か判定する。1以上であればステップS10に移る。

30

ステップS8で E_{max} が1以上でなければ、ステップS9で E_{max} が予め決めた $-K$ (K は1以上の整数)以下であるかが判定し、そうであればステップS10に移る。

【0038】

ステップS10は、 E_{max} を補正情報 Eとし、 N_F 個の各サンプルについて $E_i' = E_i - E$ により桁調整を行い、得られた E_i' を補正された E_i として使用し、整数化部12に桁調整された N_F 個のサンプルを与え、終了する。ステップS9で E_{max} が $-K$ より小でない場合も終了する。

このようにステップS8において E_{max} が1以上となることはデジタル信号サンプルXの振幅が大きく指数部Eが150を超えている場合を検出している。ステップS9において E_{max} が $-K$ 以下となることは、Xの振幅が小さく、波形の近似が悪い場合であり、例えば K は20~22程度の値に決めてもよい。

40

【0039】

このようにして桁調整された浮動小数点形式のデジタル信号サンプルを整数化部12で整数形式のデジタル信号サンプルYに変換しこの信号サンプルYの系列を圧縮部13で可逆圧縮符号化して符号列Caを生成することは第1実施形態と同様である。また、差分信号Zについても第1実施形態と同様に仮数部Mの非ゼロとなり得るビットの領域を圧縮符号化部17で符号化して符号列Cbを出力する。

復号化の際に、符号化の際の浮動小数点形式のデジタル信号サンプルXを可逆に得られるように調整情報 Eは補助符号化部33でその極性、つまり加算か減算かを含めて符号化して補助符号列Ccを生成する。

50

【0040】

図9に示した実施形態では、差分生成部14において逆桁調整部34で、整数形式のデジタル信号系列Yを、対応分割単位ごとの調整情報Eだけ逆に桁調整する。つまり桁調整部32で負の調整情報Eを加算した場合(ステップS10でEが負の場合)は逆桁調整部34において対応サンプルのデジタル信号の最上位ビットを除く23ビットをEビットだけ下位にシフトし、シフトで空いた各ビットに0を詰め、正の調整情報Eを減算した場合(ステップS10でEが正の場合)は、対応サンプルのデジタル信号の最上位ビットを除く23ビットをEビットだけ上位にシフトし、つまりEビットだけあふれさせ、下位Eビットに1を詰めて23+Eビットとする。

【0041】

この桁を逆調整した整数形式のデジタル信号サンプルYを浮動小数点化部15で浮動小数点形式のデジタル信号サンプルY'に変換し、この浮動小数点形式のデジタル信号サンプルY'と原浮動小数点形式のデジタル信号サンプルXとの差分Zを減算部16で求め、この浮動小数点形式の差分信号Zを圧縮部17で可逆圧縮符号化して符号列Cbを出力する。この場合は分割単位の符号列CaとCbと補助符号Ccが出力されることになる。この実施形態においても差分信号Zは圧縮符号化せず、端数部だけをそのまま差分情報として出力してもよい。サンプル列分割部31は図9中に破線で示すようにサンプル列分割した浮動小数点形式の信号サンプルXが桁調整部32と減算部16に供給されるように設けてもよい。

【0042】

図11は図9に示した符号化装置100と対応する復号化装置200の機能構成例を、図8の復号化装置200と対応する部分に同一参照番号を付けて示す。符号列Caは伸張部21で分割単位で可逆伸張復号化され、整数形式のデジタル信号サンプルYの系列が生成される。この実施形態では補助復号部41で補助符号Ccが復号されて調整情報Eが生成される。この調整情報Eにより整数形式のデジタル信号サンプルYに対する桁補正 $E_{\downarrow} + E$ が桁補正部42で行われる。調整情報Eが正であれば各デジタル信号サンプルYのビットを上位へEビットシフトし、Eが負であればデジタル信号サンプルYのビットを下位へEビットシフトする。この際、図9中の逆桁調整部34と同様に1又は0詰めを行う。

【0043】

この桁補正部42より整数形式のデジタル信号サンプルを浮動小数点化部22で浮動小数点形式のデジタル信号サンプルY'に変換し、これと、差分情報Cbを伸張部23で可逆伸張復号化して得た浮動小数点形式の差分信号Zと合成部24で加算合成する。この加算合成したデジタル信号を必要に応じて連結部43により連続するサンプル列に連結して再生浮動小数点形式のデジタル信号サンプルXの系列を得る。

変形例

図12に図9と対応する部分に同一参照番号を付けて示すように、整数形式のデジタル信号サンプルYに対し、逆桁調整を行うことなく、浮動小数点化部15で浮動小数点形式のデジタル信号サンプルに変換し、この変換された浮動小数点形式のデジタル信号サンプルと桁調整部32で桁調整された浮動小数点形式のデジタル信号サンプルとの差分を減算部16で求め、浮動小数点形式の差分信号Zを求めてもよい。つまり差分生成部14は原浮動小数点形式のデジタル信号サンプルXと整数形式のデジタル信号サンプルYとの差分信号を浮動小数点形式で求めればよく、図2、図9、図12に示す何れの構成によってもよい。

【0044】

図12と対応して復号化装置200では図13に示すように伸張部21で可逆伸張復号化した整数形式のデジタル信号サンプルYを、まず浮動小数点化部22で浮動小数点形式のデジタル信号サンプルに変換し、これと、伸張部23で可逆伸張復号化した浮動小数点形式の差分信号Zを合成部24で加算合成し、その後、その合成デジタル信号に対し、その指数部Eを、補助復号部41により復号された調整情報Eで桁補正部42にお

10

20

30

40

50

いて桁補正して、再生浮動小数点形式のデジタル信号サンプルXを得るようにしてもよい。

第3実施形態

上述の実施形態では、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の圧縮効率を高める場合を示した。第3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16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。

10

【0045】

この第3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。

20

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-E₀)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P₀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P₀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 x x 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 1 x x 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。

【0046】

16ビット整数値信号サンプルY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Y'の指数部Eは+8されたことになり、入力信号サンプルXの指数部Eと同じ値になる。信号サンプル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 x x 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Yの最上位の 1 より下位の整数部 x x と浮動小数点形式の信号サンプル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 0 0 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 0 0 に続く下位の全ビットは非ゼロとなり得る。

30

【0047】

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。

40

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Xが生成された場合であり、従って、サンプルXの仮数部M_xは図14Bに示すように小数点P₀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P₀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Yは

50

図14Aの場合と同じになる。従って、整数値信号サンプルYを浮動小数点形式に変換したときの仮数部 M_{γ} も図14Aの場合と同様である。

【0048】

しかしながら、差分信号の仮数部 M_{\geq} は仮数部 M_{\times} と M_{γ} の差分であり、図14Bに示すように差分信号Zの仮数部 M_{\geq} 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_{\times} において、小数点 P_{\square} 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より小さくはならない点が図14Bの場合と異なる。

【0049】

図14C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PD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 $\times\times$ から16ビット整数値信号サンプルYが図14A, 14Bと同様に生成される。従って、サンプルYを浮動小数点形式に変換したときの仮数部 M_{γ} 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_{\geq} 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。

【0050】

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のもとの信号が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_{\times} の整数部分である上位(n-8)ビット、図の例では $\times\times$ 、に対しその上位側に1を加えた整数値 $1\times\times$ を最下位ビットとする16ビット整数値信号サンプルYを構成すればよい。また、差分信号Zの仮数部 M_{\geq} は、入力信号サンプルXの仮数部Mの最上位(n-8)ビットを0とすることにより得られる。

【0051】

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2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_{\geq} 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_{\times} 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。

【0052】

図16に符号化装置の例を図2と対応する部分に同一参照符号を付けて示す。浮動小数点形式の入力信号サンプル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 付加して $16 (= 24 - 8)$ ビットの整数形式の信号サンプル Y とする。この信号サンプル Y が圧縮部 13 で可逆圧縮され、符号 C_a が出力される。

またこの 16 ビット整数形式の信号サンプル 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ンプル Y に変換される。この浮動小数点形式信号サンプル Y と入力信号サンプル 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。

【0053】

圧縮部 17 は分配部 33 と 2 つの圧縮部 17 h、17 c 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 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 17 h で、桁部分 Z_c は圧縮部 17 c でそれぞれエントロピィ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。

【0054】

従って圧縮部 17 h 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 14 B、14 C に示したように $n = 0$ 以外は圧縮部 17 h 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 17 c では、図 14 A、14 B、14 C 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 2 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。

【0055】

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 8 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。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 Y が復号される。伸張部 23 は 2 つの伸張部 23 h、23 c 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 23 h、23 c 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 23 h 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 23 c 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。

【0056】

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Yに変換される。この信号サンプル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Xが再生される。

上述において、整数形式信号サンプル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。

【0057】

図16に示す符号化装置の圧縮部17として図2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8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第2実施形態で説明した16ビットの整数値化を行うが、差分信号Zは非ゼロとなり得る2つの領域に分けず、図7で説明したように、仮数部の下位 $h = 23 - n$ ビットの符号化を行う。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4実施形態

前述の第1実施形態では、差分信号Zの各サンプルの非ゼロとなり得る下位 $h (= 23 - n)$ ビットを可逆圧縮符号化する例を示したが、差分信号Zを例えば1フレームごとにフレーム方向に抽出したビット列を可逆圧縮符号化してもよい。その符号化方法を図18を参照して以下に説明する。

【0058】

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Xの仮数部の $n (= E - E_0)$ 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7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のビットをフレーム方向に順次抽出して得たビット列 E_j を符号化を行う。

【0059】

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)のフレーム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フレーム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} としてエントロピー符号化する。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下位層化ができるので便利である。

【0060】

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 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

【0061】

図19は図18の符号化方法を図2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(図2参照)からの差分信号 Z の1フレーム分 $Z_0 \sim Z_{1023}$ が入力されると共に、対応する整数部分桁数 $n_0 \sim n_{1023}$ 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 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 $j (= 0 \sim h_{max} - 1)$ においてフレーム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

【0062】

図20は図19の圧縮符号化部17に対応する図7における差分信号復号化手段としての伸張部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(図7)からの整数桁数 n の1フレーム分 $n_0 \sim n_{1023}$ 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

【0063】

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 でフレーム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

【0064】

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 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

50

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 2 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。

【 0 0 6 5 】

入力信号サンプル 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 9 B, 1 0 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 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 $E_8 \sim E_{23}$ 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。

【 0 0 6 6 】

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。

【 0 0 6 7 】

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 2 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。

第 5 実施形態

前述の第 4 実施例ではフレーム毎に差分信号 Z の仮数部 M の端数部分の $M S B$ を同一位置にそろえて、サンプル列方向にビット列を抽出し、符号化する例を示した。第 5 実施例では、非ゼロとなり得る端数部分の $L S B$ 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、図 2 の符号化装置 1 0 0 に適用した場合の符号化装置の構成例を図 2 4 に示す。

【 0 0 6 8 】

図 2 と異なる点は圧縮符号化部 1 7 の構成である。図 2 4 に示すように、圧縮符号化部 1 7 は、端数部分抽出部 1 7 E と、フレーム内平均部 1 7 F と、減算器 1 7 G と、符号化部 1 7 D 1、1 7 D 2 とから構成されている。差分生成部 1 4 からの差分信号 Z は端数部分抽出部 1 7 E 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 1 7 F 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。

【0069】

図 25 は図 24 に対応する復号化装置の例を示す。この構成は、図 8 の復号化装置における伸張部 23 の構成を変更したものである。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。得られた差分信号 Z と、整数値信号サンプル Y の浮動小数点形式信号サンプル Y' とから、図 8 の場合と同様にして信号サンプル X が合成される。

【0070】

この第 4 実施形態は第 3 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 24 及び 25 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。

コンピュータによる実施形態

図 2、図 9、図 12、図 16、図 24 に示した符号化装置 100 は図 26 に示すような、バス 68 で互いに接続された RAM 61、CPU 62、ハードディスク 63、ROM 64、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 により機能させてもよい。

【0071】

ROM 64 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 100 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デジタル信号サンプル X の系列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 RAM 61 に読み込んで、その実行プログラムを CPU 62 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。

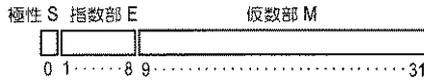
【0072】

図 8、図 11、図 13、図 17、図 26 に示した復号化装置 200 を図 24 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。

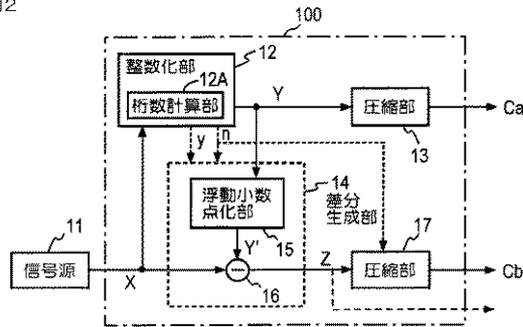
【図1】

図1



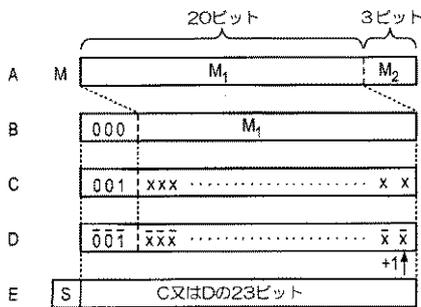
【図2】

図2



【図3】

図3



【図5】

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
$k-1$	$2^{k-1} \sim (2^k-1)$	$1 \cdot x_{23} \ x_{22} \ \dots \ x_{23-(k-2)} \cdot x_{23-(k-1)} \ \dots \ x_1$ 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

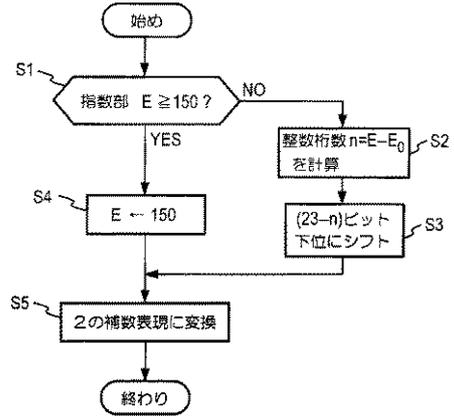
【図6】

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4】

図4



【図7】

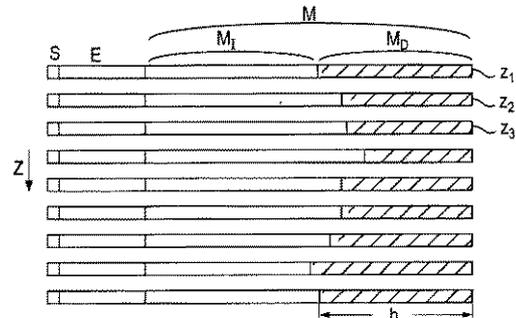


図7

【図8】

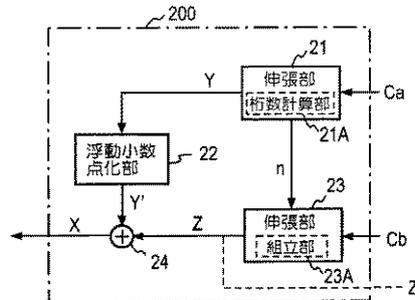
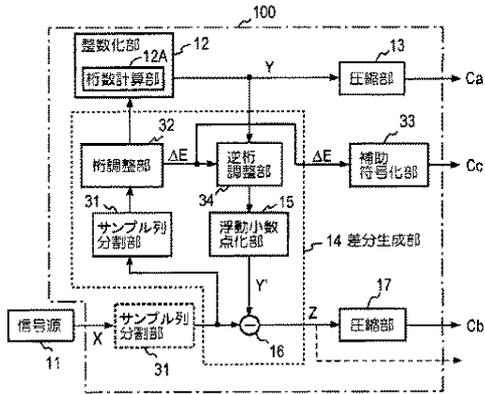


図8

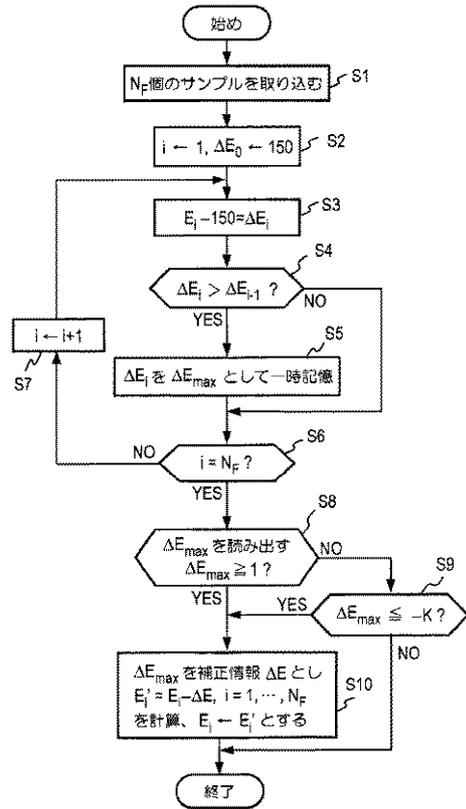
【図9】

図9



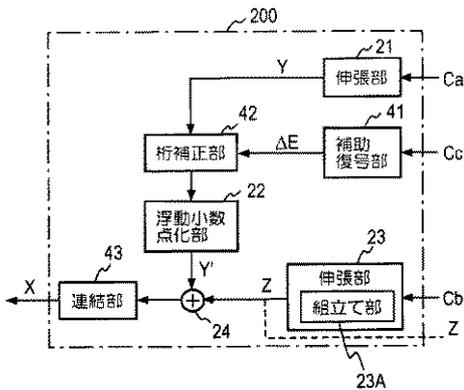
【図10】

図10



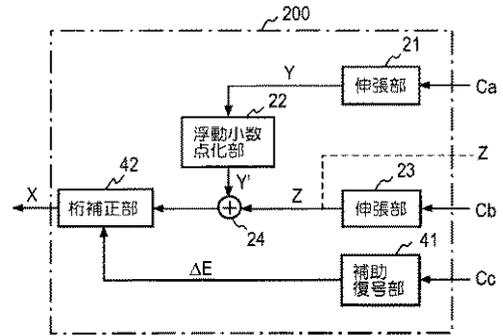
【図11】

図11



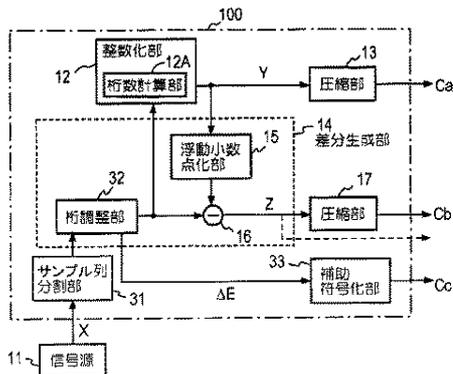
【図13】

図13



【図12】

図12



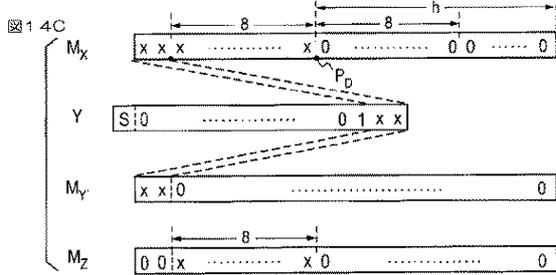
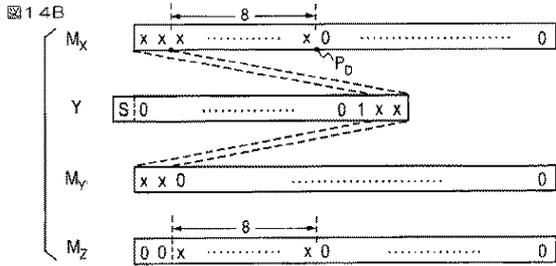
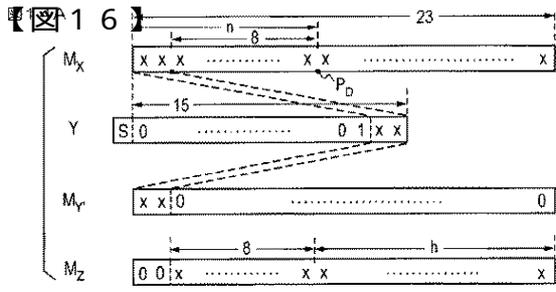


図15A

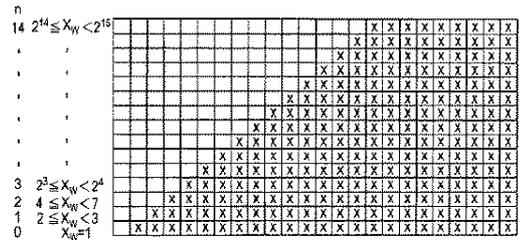


図15B

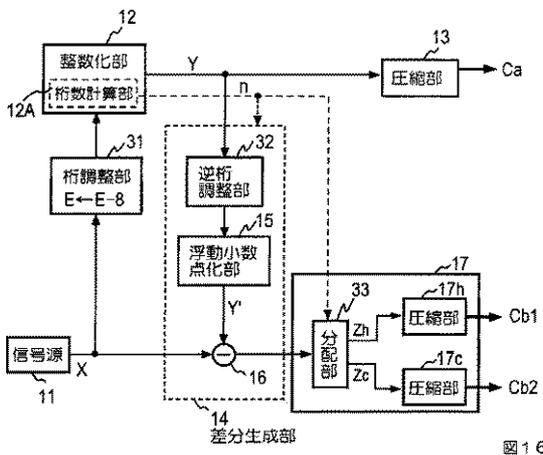
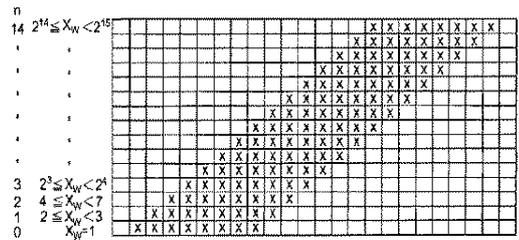


図 16

【図 17】

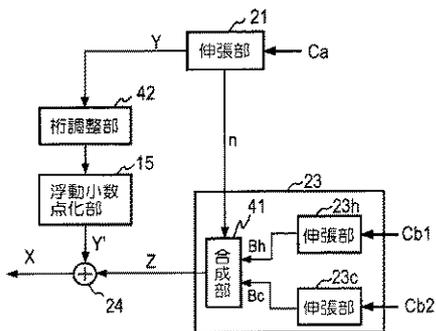


図 17

【図 18】

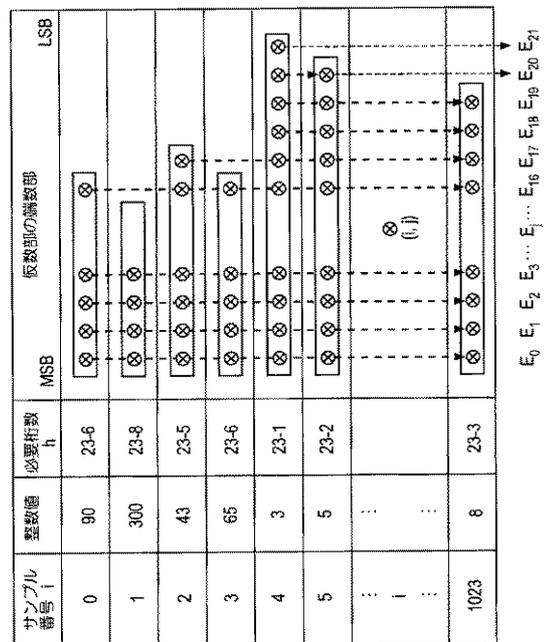
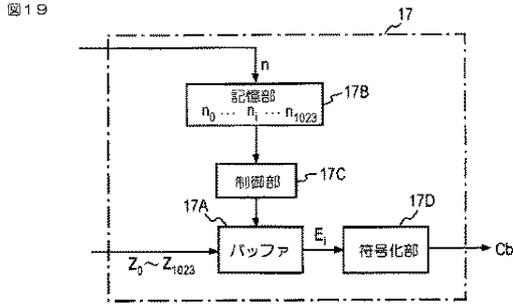
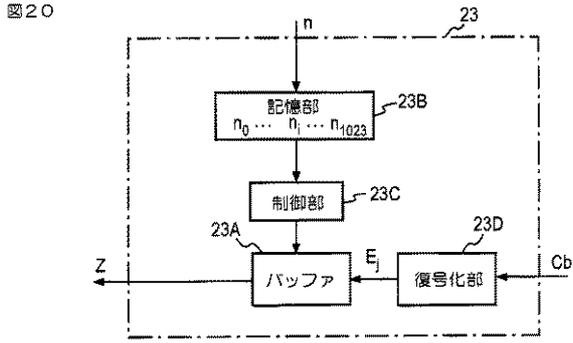


図 18

【図19】



【図20】



【図21】

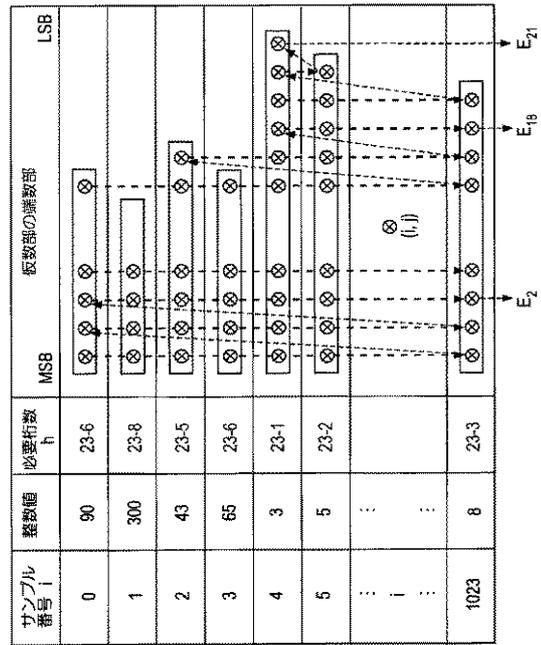


図21

【図22】

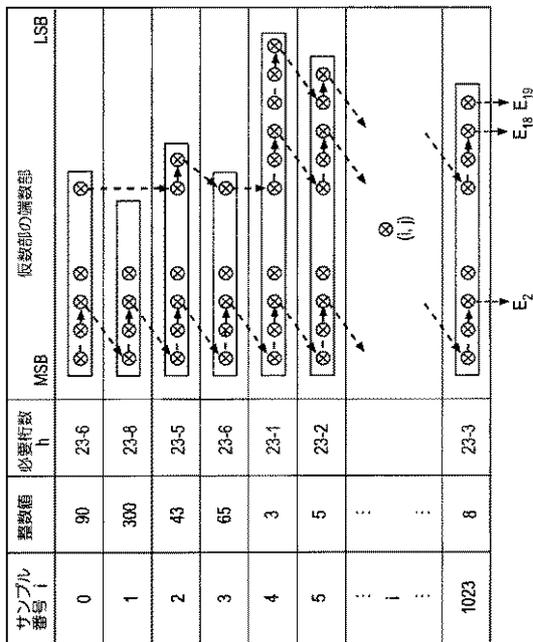


図22

【図23】

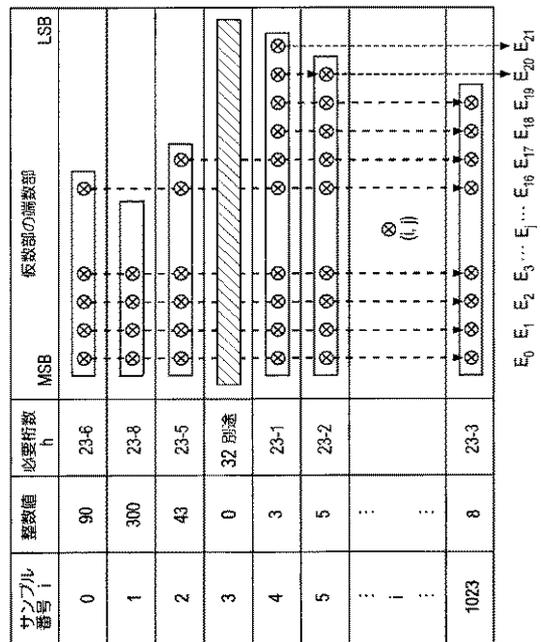


図23

【図24】

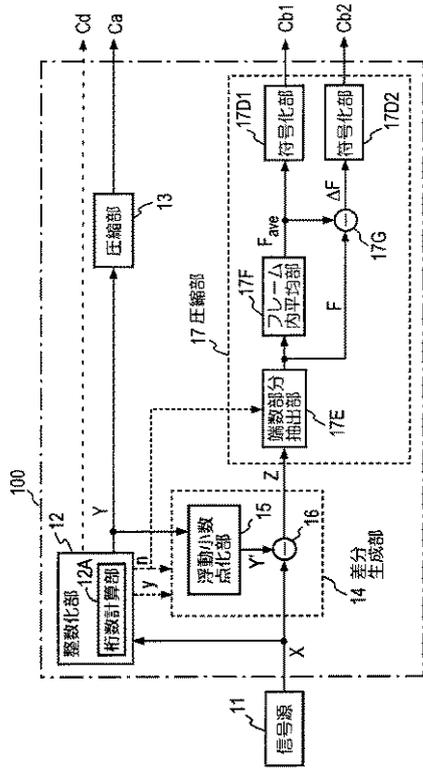


図24

【図25】

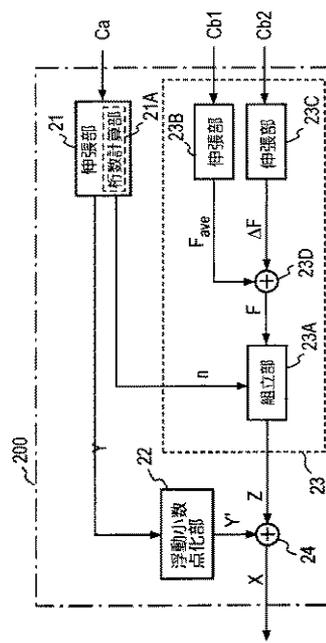
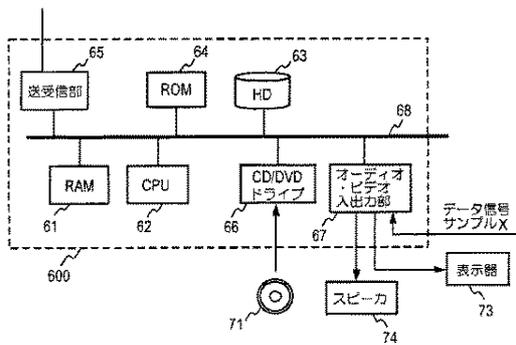


図25

【図26】

図26



フロントページの続き

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,A1)
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, 情報技術科学
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