

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3823731号
(P3823731)

(45) 発行日 平成18年9月20日(2006.9.20)

(24) 登録日 平成18年7月7日(2006.7.7)

| | | |
|-----------------------------|------------|------|
| (51) Int. Cl. | F I | |
| H03M 13/41 (2006.01) | H03M 13/41 | |
| H03M 13/45 (2006.01) | H03M 13/45 | |
| G06F 11/10 (2006.01) | G06F 11/10 | 330S |
| H03M 13/29 (2006.01) | H03M 13/29 | |
| H04L 1/00 (2006.01) | H04L 1/00 | B |
| 請求項の数 4 (全 12 頁) | | |

| | | | |
|-----------|-------------------------------|-----------|---------------------|
| (21) 出願番号 | 特願2001-8435 (P2001-8435) | (73) 特許権者 | 000005108 |
| (22) 出願日 | 平成13年1月17日 (2001.1.17) | | 株式会社日立製作所 |
| (65) 公開番号 | 特開2002-217748 (P2002-217748A) | | 東京都千代田区丸の内一丁目6番6号 |
| (43) 公開日 | 平成14年8月2日 (2002.8.2) | (74) 代理人 | 100100310 |
| 審査請求日 | 平成16年7月20日 (2004.7.20) | | 弁理士 井上 学 |
| | | (72) 発明者 | 玉木 剛 |
| | | | 東京都国分寺市東恋ヶ窪一丁目280番地 |
| | | | 株式会社日立製作所中央研究所内 |
| | | (72) 発明者 | 矢野 隆 |
| | | | 東京都国分寺市東恋ヶ窪一丁目280番地 |
| | | | 株式会社日立製作所中央研究所内 |
| | | 審査官 | 藤井 浩 |
| | | 最終頁に続く | |

(54) 【発明の名称】 誤り訂正復号器

(57) 【特許請求の範囲】

【請求項1】

タップの状態が複数あり、信号が1ビット入力される毎に前記タップの状態が変化するターボ符号器で符号化されたデータを軟判定ビタビアルゴリズム(SOVA)復号を行う復号器を有する誤り訂正復号器であって、

前記SOVA復号器は、前記ターボ符号器のタップの状態遷移をトレースバックするトレースバック処理部を有し、前記トレースバック処理部は、最も確からしい状態遷移を表す最尤パスであるかどうかを示す最尤パスフラグと、次に確からしい状態遷移を表す競合パスであるかどうかを示す競合パスフラグと、最尤パスと競合パスとの確からしさの差分に相当する尤度情報とを計算して、1ビットトレースバックする毎に硬判定値と軟判定値を確定する誤り訂正復号器。

【請求項2】

請求項1に記載の誤り訂正復号器であって、

前記トレースバック処理部は、1ビットトレースバックする毎に最尤パスから最尤パスに遷移する際に前記ターボ符号器に入力された符号を硬判定値として出力し、競合パスの集合を求め、これらの競合パスの中で前記尤度情報の最小値を軟判定値として出力することを特徴とする誤り訂正復号器。

【請求項3】

請求項1又は2の何れかに記載の誤り訂正復号器であって、

前記トレースバック処理部は、前記タップの状態数分のトレースバック計算モジュール

を有し、各計算モジュールは、前記最尤パスフラグと、前記競合パスフラグと、前記尤度情報とを計算することを特徴とする誤り訂正復号器。

【請求項 4】

請求項 1 記載の誤り訂正復号器であって、1 回のトレースバックにより、複数ビットについての硬判定値と軟判定値とを出力することを特徴とする誤り訂正復号器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、受信した符号化データの伝送誤りを訂正して復号する誤り訂正復号器に関し、特にターボ符号方式の誤り訂正復号器に関する。

【0002】

【従来の技術】

通信システムにおいて、データの伝送誤りを救済するために様々な誤り訂正符号化方式が採用されている。例えば、山口、他 1 名，“シャノン限界に迫る新しい符号化方式「ターボ符号」”，日経エレクトロニクス，1998年7月13，No.721，pp.163-177に記されているように誤り訂正能力の高い符号化方式としてターボ符号化方式が知られている。図 2 に示す通信システムでは、ターボ符号器 201 に再帰的組織畳込み符号器を二つ用意し、一つ目の畳込み符号器 204 では情報源の信号を入力順に符号化する。二つ目の畳込み符号器 206 に入力する前に、情報源の信号を一度メモリに蓄え、これをあるパターンに従った順番で取り出すインターリーバ 205 によってデータ順序を攪拌し、畳込み符号器 206 によって符号化する。前記、二つの符号化出力より符号化データ $U, Y1, Y2$ が通信路 202 に送出される。この通信路 202 を介した符号化データ $U', Y1', Y2'$ が、ターボ復号器 203 に入力され、ターボ符号の復号処理を行って復号データ U'' を復元する。

【0003】

ターボ復号器 203 は、復号器 (D1, D2) 207, 209 と、インターリーバ 208, 211 とデインターリーバ 210 を含む。復号器 (D1) 207 には、送信データ $U, Y1$ に相当する $U', Y1'$ を入力して軟判定復号を行う。また、 $Y2'$ は原信号 X をインターリーブして畳込み符号化した送信データ $Y2$ に相当し、復号器 (D1) 207 の復号データをインターリーバ 208 により、 $Y2'$ に対応するようにインターリーブして復号器 (D2) 209 に入力し、軟判定復号を行う。復号結果出力は、元のデータ順になるようにデインターリーバ 210 によってデインターリーブして復号出力 U'' を得る。この復号出力 U'' を受信データ U' として再度復号器 (D1) 207 に入力し、前述と同一の動作を繰り返す。この復号処理を複数回繰り返すことによって、ランダム発生するランダム誤りやバースト発生するバースト誤りを訂正することが可能となる。

【0004】

復号器 (D1, D2) の復号方式としては、例えば MAP (最大事後確率) 復号化方式と SOVA (軟判定ビタビアルゴリズム) 復号化方式があげられる。前者の MAP 復号化方式は、受信データの移行確率を用いて前方用確率 a と後方用確率 b とを算出し、各時間 (ビット) について前方用確率 a と後方用確率 b とを用いて “1” であるか又は “0” であるかの確率の大きい方 (硬判定復号データ) とその差 (軟判定値) とを求めるものである。

【0005】

前記 MAP 復号化方式と比較して、SOVA 復号化方式は誤り訂正の特性は劣化するものの演算量は少なく済むことが知られている。

【0006】

SOVA 復号方式は、Claude Berrou et al, “A Low Complexity Soft-Output Viterbi Decoder Architecture”, Proc. IEEE, 1993 に示されている。図 3 に SOVA 復号方式の動作概要の説明図を示す。畳込み符号器 301 は、符号器に信号が入力されるとタップの状態が遷移し、拘束長を K とすると $2^{(K-1)}$ 通りの状態が存在する。図 3 は、拘束長 $K=4$ の畳込み符号器 301 を示している。図 3 のタップの状態が、000, 001, 010, 011, 100, 101, 110, 111 となっているとき、それぞれ、State 0, State 1, State 2, State 3, State 4, State 5, State 6, State 7 と定義する。State 6 の状態で入力信号 0 が入力されると、次の状態は

10

20

30

40

50

State 7となる。このState 6の時点を $\text{Bit}=(-1)$ 、State 7の時点を $\text{Bit}=0$ とすると、入力信号0によって、State 6からState 7へ状態が遷移したことになる。この状態遷移を逆にたどることによって、入力信号系列を求めることができる。状態遷移を逆にたどるトレースバックと呼ぶ。図3では、 $\text{Bit}=0$ でState 7、 $\text{Bit}=(-1)$ でState 6、 $\text{Bit}=(-2)$ でState 4というように、復号器に入力された信号から $\text{Bit}=(-63)$ までの信号系列に対して、状態遷移の可能性を調べ、最も確からしい最尤パスを追跡することによって、 $\text{Bit}=(-63)$ から $\text{Bit}=(-62)$ に状態遷移する際に入力された信号を硬判定値として出力する。

【0007】

また、図3において、 $\text{Bit}=(-2)$ から $\text{Bit}=(-1)$ での入力信号が誤っていた場合を想定すると、別の状態遷移の可能性があるため、最尤パスの追跡の他に、 $\text{Bit}=(-1)$ から $\text{Bit}=(-2)$ で別の状態遷移について調べ、これを競合パスとして追跡する。

10

【0008】

図3において、 $\text{Bit}=(-63)$ 時点で前記最尤パスと競合パスに関して、それぞれのパスの確からしさの差分に相当する尤度情報を軟判定値として求める。

【0009】

SOVA復号化方式において、誤り訂正能力の特性を発揮するためにトレースバックするビット数 L （トレースバック長）は、拘束長 K の数倍程度必要であることが知られている。図4に、1ビットの復号結果を得るために、最低限必要なトレースバック長の値を $L_{\min}=64$ とした場合を示す。従来の技術では、 $\text{Bit}=0$ から $\text{Bit}=(-63)$ までのトレースバックを1回行うと、 $\text{Bit}=(-63)$ 時点における1ビット分の硬判定値と軟判定値が得られる。1ビット分の硬判定値と軟判定値を得るために、1回のトレースバックで L_{\min} に相当する演算量が必要で、復号する情報ビット数を N とすると、 $L_{\min} \times N$ の演算量が必要となる。

20

【0010】

【発明が解決しようとする課題】

従来の技術では、1回のSOVA復号結果を得るのに情報ビット数 N とトレースバック長 L_{\min} に比例して演算時間が大きくなるという問題がある。特に、ターボ復号処理では、復号出力を再度復号器に入力する繰り返し復号処理を行うため、伝送路の誤りが多い場合は、高い誤り訂正能力を実現するために繰り返し回数が多く必要とされる。繰り返し回数を IT とするとトータルの演算量は $IT \times L_{\min} \times N$ となり、繰り返し回数が多いほど演算時間が長くなり、信号処理に高い動作周波数が必要とされる。例えば、従来の技術によるハードウェア実装を行うと、伝送レート 384 kbit/sec のターボ復号処理に必要な動作周波数は、繰り返し回数 $IT=16$ とすると、 100 MHz にもなる。移動端末の場合は特に低消費電力化が重要な課題であるため、動作周波数を低減させることが必要となる。

30

【0011】

そこで、本発明では、SOVA復号化方式におけるトレースバック処理の演算時間を削減することを目的とする。

【0012】

【課題を解決するための手段】

前記問題を解決するために、1回のトレースバックにおいて複数ビットの硬判定出力と軟判定値を求めるトレースバック処理方式を提供する。本発明のトレースバック回路の一実施例では、図5に示すように、トレースバックにおいて、最尤パスをたどる際に1ビットごとに最尤パス方向と反対に遷移するパスを競合パスとして記憶する。これにより、1回のトレースバックで競合パス全部について調べることができるため、各ビットにおける軟判定値を求めることが可能となる。従来の技術では、1回のトレースバックを L ビット実行して軟判定値1ビット分を求めるが、本発明によればトレースバック開始点からの深さを M ビットとすると、トレースバックを M ビット実行して軟判定値1ビット分が求まる。トレースバック長が拘束長 K の数倍程度の最小値 L_{\min} を満たしていれば特性が保たれることになる。1回のトレースバックで L ビット実行すると $M \geq L_{\min}$ を満たすビットについて軟判定値が定まるため、 $L - L_{\min}$ ビット分の硬判定出力と軟判定値が1回のトレースバックで得られる。これにより、1回のトレースバックで複数ビットの復号結果が求められるた

40

50

め、情報長Nビット全部の復号結果を求めるためのトレースバック回数をN回からN/(L-L_{min})回に減らすことが可能となり、演算時間を削減できる。

【0013】

【発明の実施の形態】

図6に本発明の復号器の一実施例を示す。この復号器は、復号器の入力として受信ターボ符号データを情報長分蓄える入力信号メモリ601と、軟判定ビットピアルゴリズムによるSOVA復号器602と、前記SOVA復号化による硬判定出力と軟判定値の復号結果を出力として格納する復号結果メモリ603と、前記復号データと軟判定情報を所定回数繰り返し演算させるときにインターリーブパターンによってアドレス制御を行う制御部604と、インターリーブパターンを格納したインターリーブパターンメモリ605とを備えている。インターリーブパ

10

【0014】

図7により、データの流れを説明する。繰り返し復号奇数回目の処理では、通信路を通して受信されたターボ符号化データを格納した入力信号メモリ601より、U'をアドレス順に読み出した値をSOVA復号器602入力C0として、Y1'をアドレス順に読み出した値をC1として使用する。SOVA復号器602の出力L(U')_nは、事前尤度情報Le(U')_(n-1)と通信路値U'を減算した後、外部情報尤度Le(U')_n = $\alpha \{L(U')_n - U' - Le(U')_{(n-1)}\}$ として復号結果メモリ603にアドレス順に書き込まれる。繰り返し復号の初回は、事前尤度情報Le(U')_(n-1)を0とする。ここで α は軟判定値に対する信頼度を重み付けする係数であり、誤り訂正能力の特性に影響を与える。この α は、誤り訂正能力のビットエラーレートから通信路のノイズ状態を測定し、適応的に制御することが可能なものである。なお、この α で重み付けする箇所は図6では省略されている。次に、繰り返し復号偶数回目の処理では、入力信号メモリ601より、インターリーブ701によってインターリーブパターンに従って読み出した値をSOVA復号器入力C0として、Y2'をアドレス順に読み出した値をC1として使用する。ここでインターリーブ701の機能は、図6のインターリーブパターンメモリ605から制御部604によってインターリーブパターンを読み出すことによって、入力信号メモリ601に対するアドレスを生成することにより実現する。事前尤度情報Le(U')_(n-1)は、前回の復号で得られた外部情報尤度を復号結果メモリ603からインターリーブ702によってインターリーブパターンに従って読み出した値を使用する。インターリーブ702の機能もインターリーブ701と同様にして実現される。SOVA復号器602の出力L(U')_nは、事前尤度情報Le(U')_(n-1)と通信路値U'を減算した後、外部情報尤度Le(U')_n = $\alpha \{L(U')_n - U' - Le(U')_{(n-1)}\}$ としてデインターリーブ703によって復号結果メモリ603にインターリーブパターンに従ったアドレスで書き込む。ここで、デインターリーブ703の機能も、インターリーブ701・702と同様にして実現される。つまり、図7におけるインターリーブ701・702とデインターリーブ703は、図6において制御部604とインターリーブパターンメモリ605で表されていることになる。遅延器606は、SOVA復号器602入力C0とC2を加算したものを、SOVA復号器602の出力L(U')_nが求まるまで遅延させる回路であり、繰り返し復号最終回には0固定とする。

20

30

【0015】

次にインターリーブ701・702、デインターリーブ703の機能の実現方法に関して、図6の制御部604とインターリーブパターンメモリ605の繰り返し復号実行時の動作を、図8を用いて説明する。繰り返し復号奇数回時は、入力信号メモリの読み出しアドレス、復号結果メモリの読み出しアドレス、書き込みアドレスは1ずつインクリメントするアドレス順となるため、制御部604は信号処理のタイミングにあわせてアドレス生成を行う。繰り返し復号偶数回時は、インターリーブパターンメモリのアドレス順にインターリーブパターンを読み出した値を、入力信号メモリの読み出しアドレス、復号結果メモリの読み出しアドレス、書き込みアドレスとして使用する。制御部604で各メモリに対する信号処理タイミングにあわせてアドレスを生成する。

40

【0016】

次に、図6のSOVA復号器602について説明する。SOVA復号器602では、すべての状態遷移に

50

ついて遷移の確からしさ（メトリック値）と、遷移情報（パス値）と、ある状態に至るまでの遷移の確からしさの差分に相当する尤度情報（尤度値）とを求めるACS（Add-Compare-Select）回路607と、ACS回路607で求めたメトリック値を記憶するメトリックメモリ608と、パス値を記憶するパスメモリ102と、尤度値を記憶する尤度メモリ103と、パス値から最も確からしい遷移の軌跡を追跡するトレースバック回路101とを備えている。

【0017】

まずACS回路607における実施の形態の例を図9に示す。ACS回路607では、状態遷移の基本構造（パタフライ）に対し、遷移ブランチ毎にブランチの確からしさ b_{met} をSOVA復号器入力 C_0, C_1, C_2 の関数として求める。この関数は、符号器の構成に応じた関数となる。例えば、3GPP Release'99のターボ符号器の仕様を例とした場合、次式（1）で表される。

$$b_{met} = C_0 + C_1 + C_2 : (\text{State}[0] \wedge \text{State}[1] = 0)$$

$$= -C_0 + C_1 - C_2 : (\text{State}[0] \wedge \text{State}[1] = 1) \quad (1)$$

ここで、状態 State を二進数表記し、LSB(Least Significant Bit)の1 bit目を $\text{State}[0]$ 、2 bit目を $\text{State}[1]$ とし、 \wedge は排他論理を表すものとする。

【0018】

図6のメトリック値を格納したメトリックメモリ608より、メトリック値 MET_P_0 、 MET_P_1 をロードし、それぞれACS回路における入力側の二つの状態に対するメトリック値とする。ここで、メトリック値 MET_P_0 、 MET_P_1 は1ビット前にACS回路で計算されてメトリックメモリ608に格納されたメトリック値の内、ACS回路の入力側に対応する二つの状態に対応するメトリック値が選択されるものとする。図9では、状態0に対して入力信号0が入力された場合は状態0に遷移し、状態1に対して入力信号1が入力された場合は状態0に遷移する様子を表している。このとき、状態0に遷移する可能性としては、状態0から状態0への遷移と、状態1から状態0への遷移の二つの場合がある。状態0から状態0への遷移の確からしさAは、 $A = \text{MET_P}_0 + b_{met}$ で表され、状態1から状態0への遷移の確からしさBは、 $B = \text{MET_P}_1 - b_{met}$ で表される。この二つの遷移に対して遷移の確からしさを比較し、図9の例ではBの方がAよりも大きいため、状態1から状態0への遷移が確からしいことになる。同様に状態4に対して遷移する可能性について調べると、図9では状態1から状態4への遷移が確からしいことになる。このように、図9では状態の遷移が確からしい道筋を太線で示してある。図9の場合、遷移の確からしさBの方がAよりも大きいため、パス値を1として定義する。逆にAの方が大きい場合は、パス値を0とする。同様に、遷移の確からしさCとDを比較して、状態4に遷移するパス値をCがDよりも大きい場合は0とし、CがDより小さい場合は1とする。また、遷移の確からしさAとBの差分の絶対値を2で割った値を状態0の尤度値とし、同様にCとDの差分の絶対値を2で割った値を状態4の尤度値とする。ACS回路ですべての状態に対してメトリック値、パス値、尤度値を求め、それぞれメトリックメモリ608、パスメモリ102、尤度メモリ103に格納する。メトリック値の飽和を避けるため、1ビット前のACS回路処理で最大値をもつメトリック値を記憶しておき、各メトリック値から記憶したメトリック値を減算してからメトリックメモリ608に格納するようにするようによい。

【0019】

図1にトレースバック回路の実施例の説明図を示す。トレースバック回路101は、トレースバック開始信号 TR_LD をトリガとしてトレースバック開始時状態 TR_ST を初期状態とし、パスメモリ102から読み出したパス値と尤度メモリ103の尤度値を使って、硬判定値 SIGN と軟判定値 WGT を求めるものである。ただし、ここで $P[0]$ から $P[7]$ は、それぞれ状態0から状態7までに対するパス値を表すものとする。

【0020】

トレースバック回路101は、状態数分のトレース部104と出力選択部105から構成される。トレース部104は、最も確からしい遷移を表す最尤パスフラグ SF と、次に確からしい遷移を表す競合パスフラグ CF と、最尤パスと競合パスの確からしさの差分を表す尤度情報 W を各状態について求める回路である。各トレース部104で求められた情報は、1ビット前の

10

20

30

40

50

情報としてトレース部104にトレリス状態遷移に従ってフィードバックをかける構造になっている。例えば、図1ではBit=0の時点において、トレース部7で最尤パスフラグSF=1を求めている。この結果をBit=-1の時点においてトレース部7とトレース部6の入力としてフィードバックすることにより、トレース部7は競合パスフラグCF=1を、トレース部6は最尤パスフラグSF=1を求めている。このようにして、他のトレース部も同様な働きをすることによって、トレースバック処理が行われる。出力選択部105では、各状態のトレース部104の出力結果から、硬判定値SIGNと軟判定値WGTを計算する。

【0021】

図10にトレースバック回路のトレース部における最尤パスフラグSFを決定するアルゴリズムを示す。トレースバック開始時は、ACS回路で最もメトリック値の大きい状態をトレースバック開始時状態TR_STとし、状態TR_STに対応するトレース部の最尤パスフラグSFが1となるようにフラグを設定する。トレースバック処理中においては、1ビット前時点で最尤パスフラグSFが1であった状態から自状態に遷移してくるパスが、パスメモリから読み出したパス値と比べて正しい遷移を示していれば、自状態が最尤パスとして継承され、最尤パスフラグSFが1となる。

10

【0022】

図11にトレースバック回路のトレース部における競合パスフラグCFを決定するアルゴリズムを示す。トレースバック開始時は、全状態について競合パスは存在しないため、競合パスフラグCFを0にリセットする。また、自状態が最尤パスである場合には、競合パスとなり得ないので競合パスフラグCFは0になる。競合パスになる条件として、1ビット前において最尤パスであった状態から自状態に遷移するパスについて、パスメモリから読み出したパス値と比べて正しくない場合に、競合パスフラグCFが1となる。また、1ビット前において競合パスであった状態から自状態に遷移するパスについて、パスメモリから読み出したパス値と比べて正しい場合に、競合パスが継承されることになり、競合パスフラグCFが1となる。

20

【0023】

図12にトレースバック回路のトレース部における尤度情報Wを決定するアルゴリズムを示す。まずトレースバック開始時は、尤度情報を最大値に設定する。また、自状態が最尤パスである場合も尤度情報を最大値に設定する。1ビット前において最尤パスであった状態から自状態に遷移するパスが、パスメモリから読み出したパス値と比べて正しくない場合には、最尤パスであった状態の尤度メモリから読み出した尤度値DELTAを尤度情報の候補W_1=DELTAとして保持する。1ビット前において競合パスであった状態から自状態に遷移するパスが、パスメモリから読み出したパス値と比べて正しい場合は、競合パスの状態における1ビット前の尤度情報W(1bit前)を尤度情報の候補W_2=W(1bit前)として保持する。尤度情報の候補W_1,W_2を比較して小さい値を自状態における尤度情報W=Min(W_1,W_2)として出力する。

30

【0024】

図13にトレースバック回路の出力選択部のアルゴリズムを示す。硬判定値SIGNを求めるためには、最尤パスフラグSF=1である状態STを選択し、1ビット前の最尤パスの状態から状態STに遷移する際に符号器で入力された符号を硬判定SIGNとして出力する。パス値を符号器で入力された符号とすれば、状態STにおけるパス値が硬判定値SIGNとして出力される。次に軟判定値WGTを求めるために、競合パスフラグCF=1である状態の集合Uを求める。集合Uに属するトレース部の尤度情報Wの最小値を軟判定値WGTとして出力する。

40

【0025】

【発明の効果】

本発明によれば、情報長Nビットの復号を行う場合のトレースバックの演算量は、次式(2)のようにあらわすことができる。

$$[(N+L-L_{\min}-1)/(L-L_{\min})] \times L \times S \quad (2)$$

ただし、[x]はxを超えない最大の整数を表すものとする。Lは1回のトレースバックを行うビット数で、L_minはSOVA復号結果の軟判定値の信頼性を確保するために最低限必要な

50

トレースバックのビット数を表す。また，Sは状態数を表し，トレース部の個数に相当する。従来の技術によるトレースバックの演算量は，次式（3）で表される。

$$N \times L_{\min} \times 2 \quad (3)$$

ここで，最尤パスと競合パスの二つに対する演算量を考慮して，2倍している。

【0026】

図14に従来の技術と本発明の演算量比較を，N=512，L=64，L_{min}=32の場合を例として示す。状態数が32よりも小さい範囲では，本発明は従来の技術に比べて演算量が少なく済むことがわかる。拘束長Kが小さく，状態数が少ない場合に，本発明が有効であると言える。

【0027】

本発明により，拘束長Kが小さく状態数が少ないターボ符号において，ターボ復号器におけるSOVA復号の演算量が低減され，信号処理に必要とされるデジタル信号処理の動作周波数が低減できるという効果がある。例えば，従来の技術に比べて約1.4倍の49kGateのハードウェア実装で，伝送レート384kbit/secのターボ復号処理に必要な動作周波数は繰り返し回数IT=16として，12MHz程度となり，従来の技術の約1/8程度に抑えることができる。

【図面の簡単な説明】

【図1】本発明におけるターボ復号方式実施例のトレースバック回路の説明図。

【図2】ターボ符号を用いた通信システムの説明図。

【図3】ターボ復号におけるトレースバックの説明図。

【図4】従来技術におけるトレースバックの問題点の説明図。

【図5】本発明におけるトレースバック方式の説明図。

【図6】本発明におけるターボ復号器の構成図。

【図7】本発明におけるターボ繰り返し復号時の動作説明図。

【図8】本発明におけるターボ復号器制御部の動作説明図。

【図9】本発明におけるターボ復号器ACS回路の説明図。

【図10】トレースバック回路における最尤パスフラグ決定アルゴリズム。

【図11】トレースバック回路における競合パスフラグ決定アルゴリズム。

【図12】トレースバック回路における尤度情報決定アルゴリズム。

【図13】トレースバック回路における硬判定値・軟判定値決定アルゴリズム。

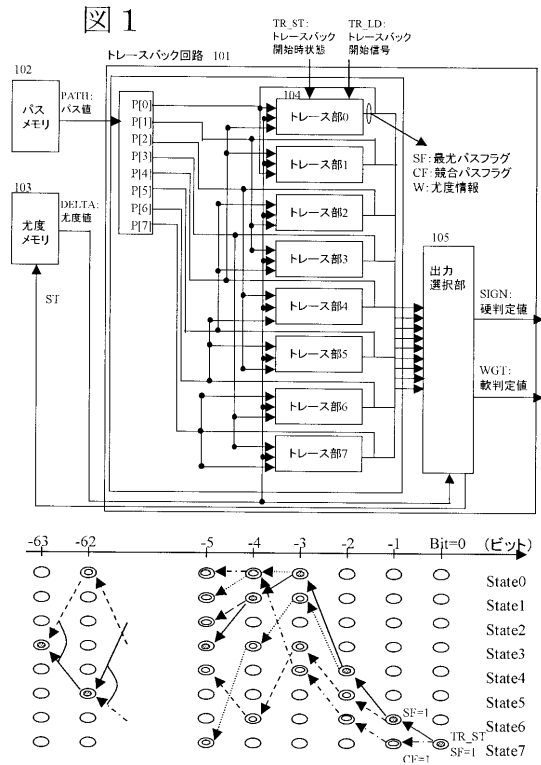
【図14】従来の復号処理と本発明の復号処理との演算量を比較したグラフ。

【符号の説明】

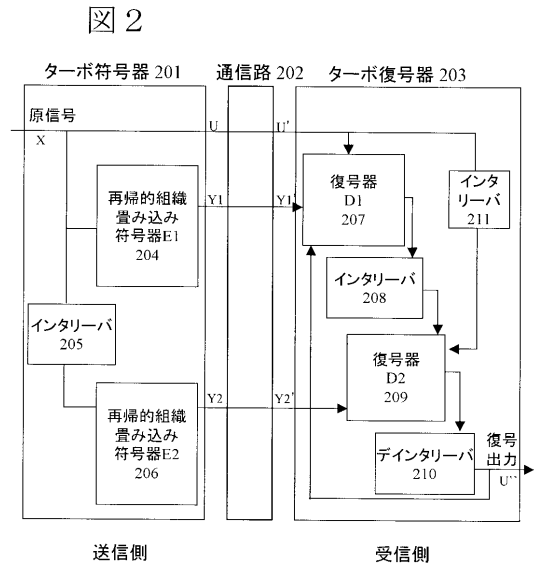
101・・・トレースバック回路，102・・・バスメモリ，103・・・尤度メモリ，104・・・トレース部，105・・・出力選択部，201・・・ターボ符号器，202・・・通信路，203・・・ターボ復号器，204，206・・・再帰的組織畳込み符号器，205，208，211・・・インタリーバ，207，209・・・復号器，210・・・デインタリーバ，301・・・畳込み符号器，601・・・入力信号メモリ，602・・・SOVA復号器，603・・・復号結果メモリ，604・・・制御部，605・・・インタリーブパターンメモリ，606・・・遅延器，607・・・ACS回路，608・・・メトリックメモリ3，701，702・・・インタリーバ，703・・・デインタリーバ

。

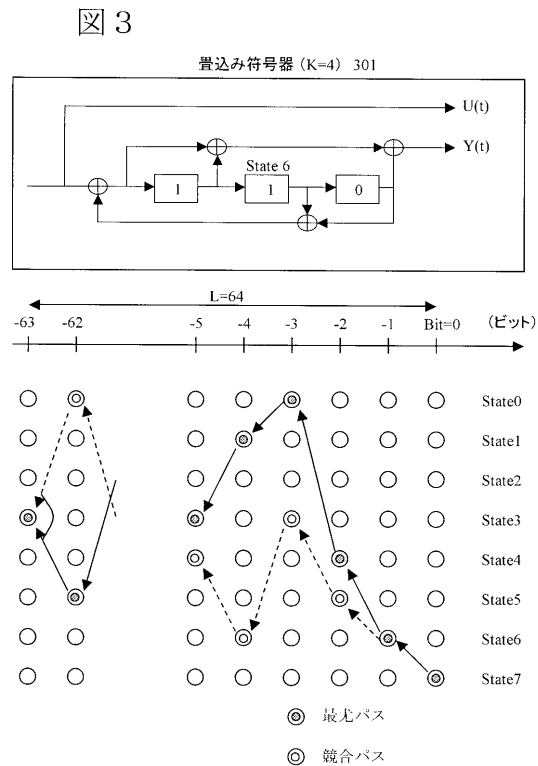
【図 1】



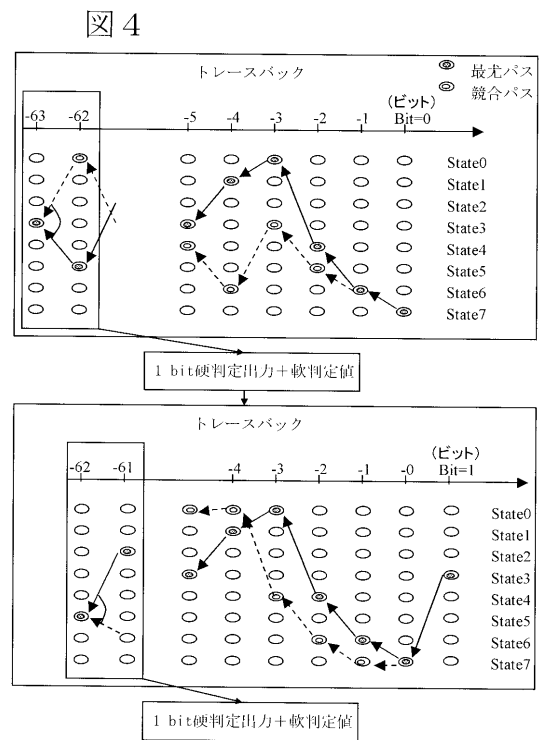
【図 2】



【図 3】

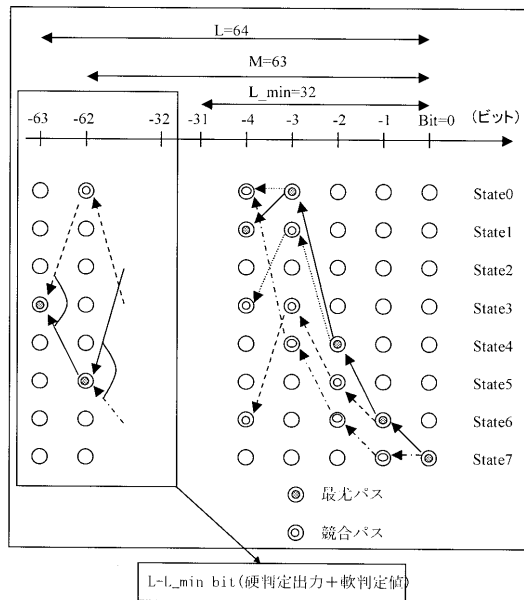


【図 4】



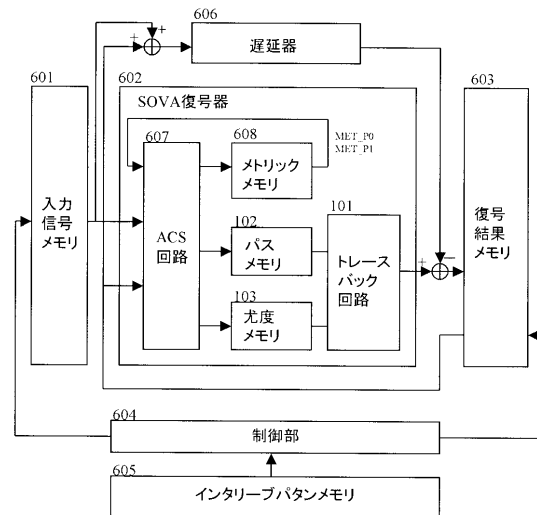
【図 5】

図 5



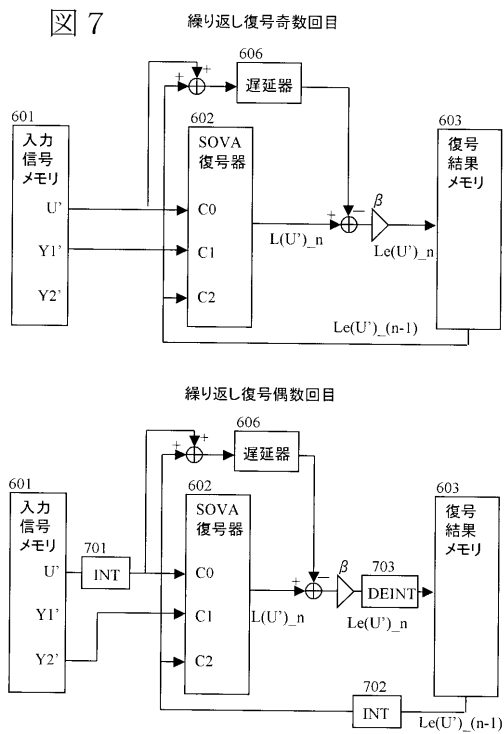
【図 6】

図 6



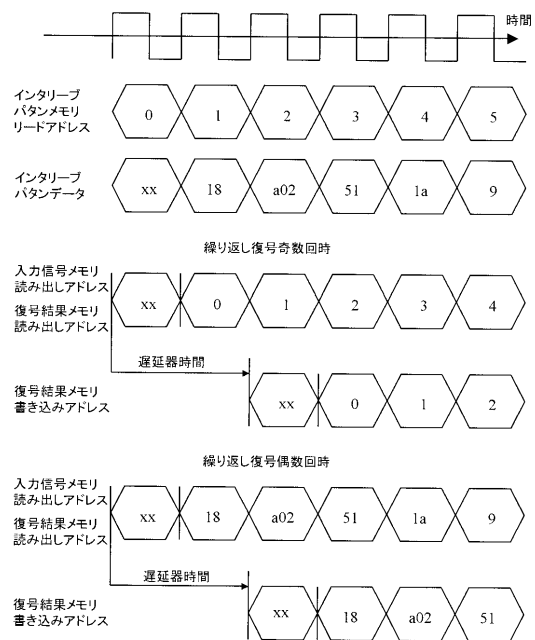
【図 7】

図 7



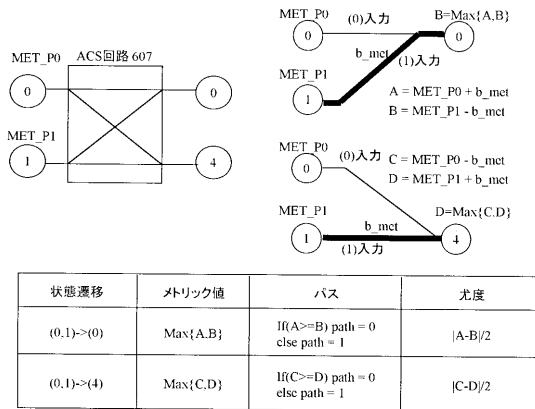
【図 8】

図 8



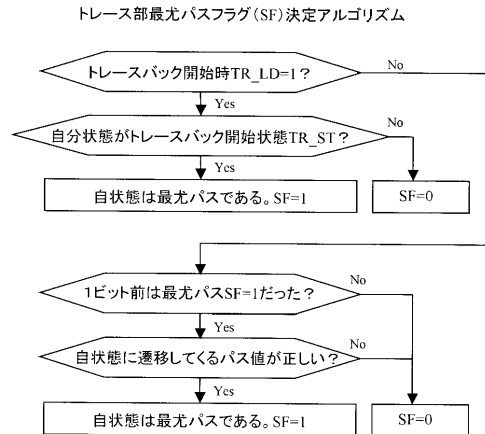
【図 9】

図 9



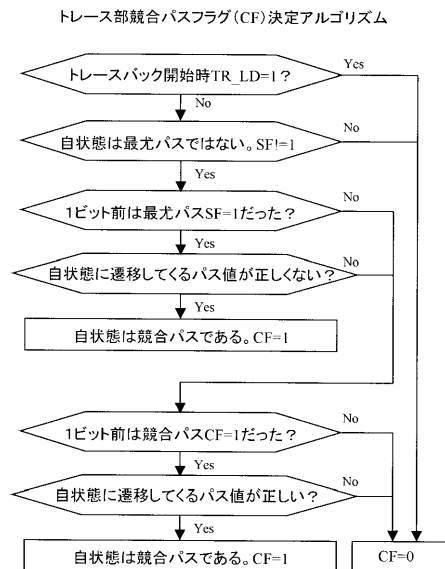
【図 10】

図 10



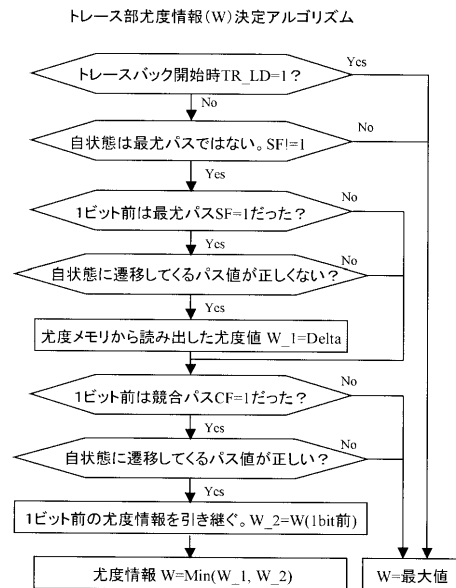
【図 11】

図 11



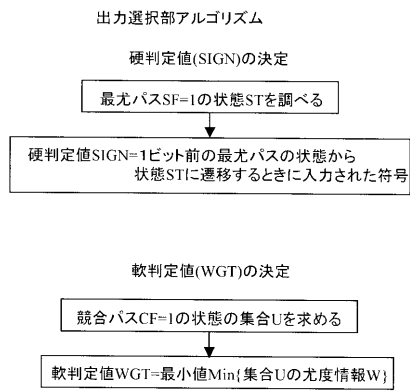
【図 12】

図 12



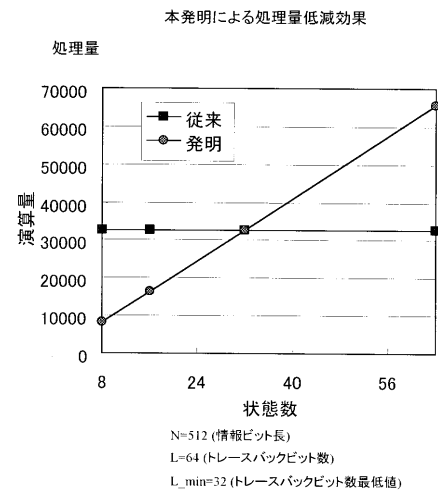
【図 1 3】

図 1 3



【図 1 4】

図 1 4



フロントページの続き

(56)参考文献 特開平 1 1 - 1 6 3 7 4 2 (J P , A)
特開 2 0 0 0 - 2 5 2 8 4 0 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

H03M 13/00 - 13/53

G06F 11/10

H04L 1/00

G11B 20/18