(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2008-311869
(P2008-311869A)
(43)公開日　平成20年12月25日(2008.12.25)

(51)Int.Cl.			F I			テーマコード（参考）
H04B	1/707	(2006.01)	H04J	13/00	D	5K022
H04Q	7/38	(2006.01)	H04B	7/26	109M	5K067

審査請求 未請求　請求項の数 18　ＯＬ　（全 22 頁）

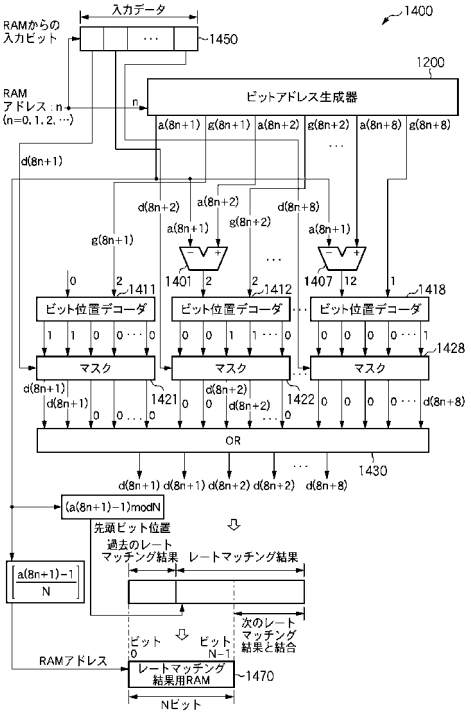
(21)出願番号　特願2007-156629 (P2007-156629)
(22)出願日　平成19年6月13日 (2007.6.13)

(71)出願人　598036300
テレフオンアクチーボラゲット　エル　エム　エリクソン（パブル）
スウェーデン国　ストックホルム　エス－１６４　８３
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治

最終頁に続く

(54)【発明の名称】レートマッチング装置及びその方法、デレートマッチング装置及びその方法、通信装置、プログラム、記録媒体

(57)【要約】
【課題】　レートマッチング／デレートマッチング処理をさらに高速に実行するための技術を提供する。
【解決手段】　入力データに含まれる各ビットの出力数を決定し、該入力データのデータ長を伸縮するレートマッチング装置は、前記入力データのm番目のビットを出力する出力回数ｇ(m)を、前記入力データのそれぞれ異なるビットについて、該入力データの伸縮を行うための基本パラメータと、該mの値と、に基づいてそれぞれ決定する複数の決定手段と、前記複数の決定手段のそれぞれに対して、該決定手段が処理すべき前記入力データのビットの前記入力データにおける順番を割り当てる割当手段と、前記割当手段が前記複数の決定手段にそれぞれ割り当てた前記順番のビットを、当該ビットについて前記決定手段が決定した前記出力回数出力する出力手段と、を備える。
【選択図】　図１４



【特許請求の範囲】
【請求項１】
入力データに含まれる各ビットの出力数を決定し、該入力データのデータ長を伸縮するレートマッチング装置であって、
前記入力データのｍ番目のビットを出力する出力回数ｇ(ｍ)を、前記入力データのそれぞれ異なるビットについて、該入力データの伸縮を行うための基本パラメータと、該ｍの値と、に基づいてそれぞれ決定する複数の決定手段と、
前記複数の決定手段のそれぞれに対して、該決定手段が処理すべき前記入力データのビットの前記入力データにおける順番を割り当てる割当手段と、
前記割当手段が前記複数の決定手段にそれぞれ割り当てた前記順番のビットを、当該ビットについて前記決定手段が決定した前記出力回数出力する出力手段と、
を備えることを特徴とするレートマッチング装置。
【請求項２】
前記基本パラメータは、レートマッチング・アルゴリズムにおける初期値e_{ini}（≧１）、減分値e_{minus}、増分値e_{plus}（≧e_{ini}）であり、
前記複数の決定手段のそれぞれは、
前記初期値e_{ini}から前記減分値e_{minus}をｍ回引き算した値を正にするために、当該値に対して加算すべき前記増分値e_{plus}の個数の最小値ｑ(ｍ)を計算する計算手段と、
ｑ(ｍ)－ｑ(ｍ－１)に基づいて前記入力データのｍ番目のビットを出力する出力回数ｇ(ｍ)を決定する回数決定手段と、
を備えることを特徴とする請求項１に記載のレートマッチング装置。
【請求項３】
前記計算手段は、前記ｑ(ｍ)を

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

により計算することを特徴とする請求項２に記載のレートマッチング装置。
【請求項４】
前記回数決定手段は、前記回数ｇ(ｍ)を、
前記入力データのビットを挿入する場合は、
ｇ(ｍ)＝ｑ(ｍ)－ｑ(ｍ－１)＋１
により決定し、
前記入力データのビットを削除する場合は、
ｇ(ｍ)＝１－(ｑ(ｍ)－ｑ(ｍ－１))
により決定する
ことを特徴とする請求項２又は３に記載のレートマッチング装置。
【請求項５】
前記複数の決定手段のそれぞれは、
当該決定手段が処理すべき前記入力データのｍ番目のビットについての、前記出力手段による出力結果が、伸縮済みデータにおいて占めるべき箇所の先頭ビット位置ａ(ｍ)を、前記基本パラメータと、該ｍの値とに基づいて決定するビット位置決定手段を更に備え、
前記出力手段は、
前記入力データに含まれるｍ番目のビットを、前記ビット位置決定手段において決定された前記先頭ビット位置ａ(ｍ)に対応するアドレスから、前記回数決定手段において決定された前記回数ｇ(ｍ)の個数、メモリに格納する格納手段を備える
ことを特徴とする請求項２乃至４のいずれか１項に記載のレートマッチング装置。
【請求項６】
前記ビット位置決定手段は、前記先頭ビット位置ａ(ｍ)を、
前記入力データのビットを挿入する場合は、
ａ(ｍ)＝ｍ＋ｑ(ｍ－１)

により決定し、
前記入力データのビットを削除する場合は、
$a(m)=m-q(m-1)$
により決定する
ことを特徴とする請求項５に記載のレートマッチング装置。
【請求項７】
データ長が伸縮された伸縮済みデータを伸縮前データに復元するデレートマッチング装置であって、
前記伸縮前データのｍ番目のビットの、前記伸縮済みデータにおける対応箇所の先頭ビット位置ａ(ｍ)を、前記伸縮前データのそれぞれ異なるビットについて、該伸縮前データが伸縮されるときに用いられた基本パラメータと、該ｍの値と、に基づいてそれぞれ決定する複数の決定手段と、
前記複数の決定手段のそれぞれに対して、当該決定手段が処理すべきビットの前記伸縮前データにおける順番を割り当てる割当手段と、
前記割当手段が前記複数の決定手段にそれぞれ割り当てた前記順番のビットについて、該決定手段がそれぞれ決定した前記先頭ビット位置から開始するビットを用いて、前記伸縮前データのビットを出力する出力手段と、
を備えることを特徴とするデレートマッチング装置。
【請求項８】
前記基本パラメータは、レートマッチング・アルゴリズムにおける初期値e_{ini}（≧１）、減分値e_{minus}、増分値e_{plus}（≧e_{ini}）であり、
前記複数の決定手段のそれぞれは、
前記初期値e_{ini}から前記減分値e_{minus}をｍ回引き算した値を正にするために、当該値に対して加算すべき前記増分値e_{plus}の個数の最小値ｑ(ｍ)を計算する計算手段と、
ｑ(ｍ－１)と、前記ｍとに基づいて、前記先頭ビット位置ａ(ｍ)を決定する位置決定手段と、
を備えることを特徴とする請求項７に記載のデレートマッチング装置。
【請求項９】
前記計算手段は、前記ｑ(ｍ)を

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

により計算することを特徴とする請求項８に記載のデレートマッチング装置。
【請求項１０】
前記位置決定手段は、前記先頭ビット位置ａ(ｍ)を、
伸長されたデータを復号する場合は、
$a(m)=m+q(m-1)$
により決定し、
収縮されたデータを復号する場合は、
$a(m)=m-q(m-1)$
により決定する
ことを特徴とする請求項８又は９に記載のデレートマッチング装置。
【請求項１１】
前記複数の決定手段のそれぞれは、
当該決定手段が処理すべき前記伸縮前データのｍ番目のビットについて、
前記伸縮前データから前記伸縮済みデータが生成される際に該伸縮前データのｍ番目のビットが出力された回数ｇ(ｍ)を、前記基本パラメータと、該ｍの値と、に基づいて決定する回数決定手段を備え、
前記出力手段は、
前記割当手段が前記複数の決定手段にそれぞれ割り当てた前記順番のビットについて、

該決定手段がそれぞれ決定した前記先頭ビット位置から、前記回数決定手段において決定された前記回数個分の前記伸縮済みデータのビットを読み出し、当該回数個分の前記伸縮済みデータのビットの軟判定値の和を、前記伸縮前データのビットの軟判定値として出力する
ことを特徴とする請求項８乃至１０のいずれか１項に記載のデレートマッチング装置。
【請求項１２】
前記回数決定手段は、ｑ(ｍ)－ｑ(ｍ－１)に基づいて前記回数ｇ(ｍ)を決定する
ことを特徴とする請求項１１に記載のデレートマッチング装置。
【請求項１３】
前記回数決定手段は、前記回数ｇ(ｍ)を、
伸長されたデータを復号する場合は、
ｇ(ｍ)＝ｑ(ｍ)－ｑ(ｍ－１)＋１
により決定し、
収縮されたデータを復号する場合は、
ｇ(ｍ)＝１－(ｑ(ｍ)－ｑ(ｍ－１))
により決定する
ことを特徴とする請求項１２に記載のデレートマッチング装置。
【請求項１４】
請求項１乃至６のいずれか１項に記載のレートマッチング装置と、請求項７乃至１３のいずれか１項に記載のデレートマッチング装置と、の少なくともいずれかを備えることを特徴とする通信装置。
【請求項１５】
入力データに含まれるビットの出力数を決定し、該入力データのデータ長を伸縮するレートマッチング装置におけるレートマッチング方法であって、
前記入力データのｍ番目のビットを出力する出力回数ｇ(ｍ)を、前記入力データのそれぞれ異なるビットについて、該入力データの伸縮を行うための基本パラメータと、該ｍの値と、に基づいてそれぞれ並列的に決定する決定工程と、
前記入力データに含まれる各ビットを、当該ビットについて前記決定工程において決定された前記出力回数出力する出力工程と、
を備えることを特徴とするレートマッチング方法。
【請求項１６】
データ長が伸縮された伸縮済みデータを伸縮前データに復元するデレートマッチング装置におけるデレートマッチング方法であって、
前記伸縮前データのｍ番目のビットの、前記伸縮済みデータにおける対応箇所の先頭ビット位置ａ(ｍ)を、前記伸縮前データのそれぞれ異なるビットについて、該伸縮前データが伸縮されるときに用いられた基本パラメータと、該ｍの値と、に基づいてそれぞれ並列的に決定する決定工程と、
前記決定工程においてそれぞれ決定された前記先頭ビット位置から開始する前記伸縮済みデータのビットを用いて、前記伸縮前データのビットを出力する出力工程と、
を備えることを特徴とするデレートマッチング方法。
【請求項１７】
コンピュータを、請求項１乃至６のいずれか１項に記載のレートマッチング装置、又は、請求項７乃至１３のいずれか１項に記載のデレートマッチング装置として機能させるためのプログラム。
【請求項１８】
請求項１７に記載のプログラムを格納したコンピュータで読み取り可能な記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明はレートマッチング装置及びその方法、デレートマッチング装置及びその方法、

通信装置、プログラム、記録媒体に関し、特に、レートマッチング処理／デレートマッチング処理を高速化する技術に関する。

【背景技術】

【０００２】

ＷＣＤＭＡ（Wideband Code Division Multiple Access、広帯域符号分割多重アクセス方式）通信システムでは、通信装置間でデータを符号化して送受信する際に、チャネル符号化後の各ビットに対してレートマッチング処理が実行される（非特許文献１）。レートマッチングは、入力データのビット一部を繰り返したり（レピテション：Repetition、挿入）間引いたり（パンクチャ：Puncture、削除）することによって入力データのデータ長を伸縮する処理である。ＷＣＤＭＡ通信システムにおいて符号化データを送信する場合、トランスポート・チャネルを同一の物理チャネルに多重化するが、一般に通信品質の要求条件はトランスポート・チャネル毎に異なる。このため、ＷＣＤＭＡ通信システムでは、レートマッチングを行うことで誤り訂正の強度を微調整することによって、全てのトランスポート・チャネルが同時に過不足なく目標品質を達成できるようにしている。

【０００３】

図１は、３ＧＰＰリリース６のレートマッチング処理のアルゴリズムを示している（非特許文献１）。図１において、ｅはレートマッチング処理対象のデータの各ビットについてレピテション又はパンクチャの有無を決定するためのパラメータ変数である。ｍは処理対象のデータのカレントビット（処理対象ビット）のインデックスを示すパラメータ変数である。

【０００４】

X_iは処理対象データのデータ長であり、$x_{i,m}$は処理対象データのカレントビットのビット値である。e_{ini}はパラメータｅの初期値であり、e_{minus}、e_{plus}はパラメータｅの更新値（e_{minus}は減分値、e_{plus}は増分値）である。e_{ini}、e_{minus}、e_{plus}は、データ長X_i等に基づいて算出されるが、その算出手順の詳細は３ＧＰＰリリース６の標準仕様に定められている（非特許文献１）。なお、e_{ini}、e_{plus}には$1 \leqq e_{ini} \leqq e_{plus}$の関係がある。

【０００５】

レートマッチング処理の一例として、データ長が３２ビットのデータ（$X_i = 32$）をパンクチャして２６ビットにする場合を説明する。処理の条件を$e_{ini} = 1$、$e_{minus} = 6$（パンクチャされるビット数）、$e_{plus} = 32$（処理前のデータのビット数）とする。この場合、ｍ＝１～３２について、「ｅ＝ｅ－e_{minus}」を実行した直後のｅの値を図２に示す。図２のように、ｍ＝１、６、１１、１７、２２、２７について（ｅ－e_{minus}）≦０となるため（斜線部）、１、６、１１、１７、２２、２７ビット目の６ビットがパンクチャされる。

【０００６】

レートマッチング処理は符号化された全てのビットに対して行われるが、ｅＨＳＰＡ（Evolved High Speed Packet Access）技術やＬＴＥ（Long Term Evolution）技術等により、今後データ転送レートが飛躍的に増大することが見込まれている。しかし、上記のように、あるビットのパラメータｅはその直前のビットのパラメータｅに基づいて算出されるため、図１のアルゴリズムに従って処理を進めると先頭ビットから順に各ビットについて逐次的に演算する必要があり、単に処理のクロックサイクルを挙げるだけでは十分な処理速度を得ることができない。このため、レートマッチング／デレートマッチングの処理を高速化する仕組みが必要である。

【０００７】

レートマッチングを高速化するための技術としては、処理対象のデータを複数のブロックに分割し、各ビットのパラメータｅをブロック毎に並列的に算出して、パンクチャ／レピテションするビットを決定する構成が知られている（特許文献１）。この構成によれば、パラメータｅをブロック毎に並列的に算出するため、レートマッチング／デレートマッチング処理を高速化することができる。

【非特許文献１】Universal Mobile Telecommunications System (UMTS); Multiplexing and channel coding (FDD) (3GPP TS 25.212 version 6.10.0 Release 6)
【特許文献１】特開２００２－１９９０４８号公報
【発明の開示】
【発明が解決しようとする課題】
【０００８】
しかしながら、特許文献１に開示された構成においても、各ブロック内ではその先頭ビットのパラメータｅに基づいて逐次的にパラメータｅを算出する必要があるため、この手法に基づく処理の高速化には限界がある。
【０００９】
本発明は上記課題に鑑みなされたものであり、レートマッチング／デレートマッチング処理をさらに高速に実行するための技術を提供することを目的とする。
【課題を解決するための手段】
【００１０】
上記目的を達成するため、本発明によれば、
入力データに含まれる各ビットの出力数を決定し、該入力データのデータ長を伸縮するレートマッチング装置であって、
前記入力データのｍ番目のビットを出力する出力回数ｇ(ｍ)を、前記入力データのそれぞれ異なるビットについて、該入力データの伸縮を行うための基本パラメータと、該ｍの値と、に基づいてそれぞれ決定する複数の決定手段と、
前記複数の決定手段のそれぞれに対して、該決定手段が処理すべき前記入力データのビットの前記入力データにおける順番を割り当てる割当手段と、
前記割当手段が前記複数の決定手段にそれぞれ割り当てた前記順番のビットを、当該ビットについて前記決定手段が決定した前記出力回数出力する出力手段と、
を備えることを特徴とするレートマッチング装置が提供される。
【００１１】
また、本発明によれば、
データ長が伸縮された伸縮済みデータを伸縮前データに復元するデレートマッチング装置であって、
前記伸縮前データのｍ番目のビットの、前記伸縮済みデータにおける対応箇所の先頭ビット位置ａ(ｍ)を、前記伸縮前データのそれぞれ異なるビットについて、該伸縮前データが伸縮されるときに用いられた基本パラメータと、該ｍの値と、に基づいてそれぞれ決定する複数の決定手段と、
前記複数の決定手段のそれぞれに対して、当該決定手段が処理すべきビットの前記伸縮前データにおける順番を割り当てる割当手段と、
前記割当手段が前記複数の決定手段にそれぞれ割り当てた前記順番のビットについて、該決定手段がそれぞれ決定した前記先頭ビット位置から開始するビットを用いて、前記伸縮前データのビットを出力する出力手段と、
を備えることを特徴とするデレートマッチング装置が提供される。
【００１２】
また、本発明によれば、上記のレートマッチング装置と、上記のデレートマッチング装置との少なくともいずれかを備えることを特徴とする通信装置が提供される。
【発明の効果】
【００１３】
本発明によれば、レートマッチング／デレートマッチング処理をさらに高速に実行するための技術を提供することができる。
【発明を実施するための最良の形態】
【００１４】
以下、添付図面を参照して本発明に係る実施の形態を詳細に説明する。ただし、この実施の形態に記載されている構成要素はあくまでも例示であり、本発明の範囲をそれらのみ

に限定する趣旨のものではない。また、本実施の形態で説明されている特徴の組み合わせの全てが発明の解決手段に必須のものとは限らない。

【００１５】

（通信装置の構成）

まず、レートマッチング／デレートマッチング処理を実行する通信装置の機能構成について、図３を参照して説明する。図３は、通信装置３００の機能構成の概略を示すブロック図である。なお、通信装置３００には、ＷＣＤＭＡ通信を行うアンテナ装置、当該アンテナ装置を内蔵又は外付けした携帯電話機、パーソナルコンピュータ等の移動局装置、及び基地局装置が含まれる。

【００１６】

図３のように、通信装置３００は、データを送信するための機能要素として、上位アプリケーション処理部３０２、レートマッチング部３０６が含まれるチャネル符号化部３０４、チャネル多重化部３０８、物理チャネル・マッピング部３１０、拡散部３１２、変調部３１４、送信部３１６を備える。また、データを受信するための機能要素として、通信装置３００は、受信部３２２、復調部３２４、逆拡散部３２６、トランスポートチャネル・マッピング部３２８、チャネル分離部３３０、デレートマッチング部３３４が含まれるチャネル復号部３３２、上位アプリケーション処理部３３６を備える。

【００１７】

データを送信する場合は、まず、上位アプリケーション処理部３０２からチャネル符号化部３０４へ、各トランスポート・チャネルのトランスポート・ブロックが入力される。チャネル符号化部３０４では、トランスポート・チャネル毎にチャネル符号化処理が行われる。チャネル符号化部３０４で実行される処理には、以下の処理が含まれる。

・トランスポート・ブロックに対するＣＲＣ（Cyclic Redundancy Check、巡回冗長検査）ビットの付加。
・トランスポート・ブロックの結合、及び符号ブロックの分割。
・誤り訂正符号（畳み込み符号又はターボ符号）による符号化。
・レートマッチング部３０６によるレートマッチング。
・誤り訂正効果を高めるための第１インタリーブ。
・無線フレーム分のデータへの分割。

なお、下り回線、即ち、通信装置３００が基地局装置として移動局装置へデータを送信する場合は、ＤＴＸ（Discontinuous Transmission、不連続送信）を行うビット位置に第１ＤＴＸ表示を挿入する処理も実行される。

【００１８】

チャネル符号化部３０４でトランスポート・チャネル毎にチャネル符号化されたトランスポート・ブロックは、チャネル多重化部３０８において無線フレーム毎につなぎ合わされて多重化される。そして、物理チャネル・マッピング部３１０において、物理チャネル毎に分割されて、１フレーム分のデータが、個別物理データ・チャネルのスロットにマッピングされる。なお、物理チャネル・マッピング部３１０では、各無線フレーム内で第２インタリーブが実行される。また、下り回線の場合は、さらに第２ＤＴＸ表示が挿入される。

【００１９】

物理チャネルへマッピングされたデータは、拡散部３１２においてチャネライゼーション符号、及びスクランブリング符号により拡散され、変調部３１４において変調されて、送信部３１６から電波信号として送信される。

【００２０】

一方、データを受信する場合は、受信部３２２において電波信号が受信されると、復調部３２４において復調される。そして、逆拡散部３２６においてスクランブリング符号、及びチャネライゼーション符号による逆拡散が行われ、トランスポートチャネル・マッピング部３２８へ入力される。

【００２１】

トランスポートチャネル・マッピング部３２８では、個別物理データ・チャネルのスロットから無線フレームが復元されて、第２インタリーブを元に戻すデインタリーブが行われる。また、下り回線、即ち、通信装置３００が移動局装置として基地局装置からデータを受信した場合は第２ＤＴＸ表示を削除する処理も行われる。

【００２２】

次に、チャネル分離部３３０においてトランスポート・チャネル毎のデータへ分離され、チャネル復号部３３２においてトランスポート・チャネル毎にチャネル復号処理が行われる。チャネル復号部３３２で実行される処理には、以下の処理が含まれる。

・無線フレーム分割されたデータの結合。
・第１インタリーブを元に戻すためのデインタリーブ
・データ長が伸縮されたデータを元に戻すためのデレートマッチング部３３４によるデレートマッチング。
・誤り訂正符号（畳み込み符号又はターボ符号）による復号。
・ＣＲＣビットの抽出、チェック。

なお、下り回線、即ち、通信装置３００が移動局装置として基地局装置からデータを受信した場合は、第１ＤＴＸ表示を削除する処理も実行される。

【００２３】

チャネル復号部３３２においてチャネル復号されたデータは上位アプリケーション処理部３３６へ渡される。

【００２４】

（レートマッチング回路の構成例）

本実施形態に係るレートマッチング装置、デレートマッチング装置は、図３のレートマッチング部３０６、デレートマッチング部３３４に適用される。ここで、本実施形態に係るレートマッチング装置、デレートマッチング装置の利点をより一層明確にするために、本実施形態に係る構成の詳細な説明に先だって、図１のアルゴリズムに対応する回路とその問題点について概説しておく。

【００２５】

図１のアルゴリズムを実行する電子回路は、例えば図４のように構成することができる。図４は、レートマッチングを行う回路４００の一例を示す図である。図４のように、回路４００は、マルチプレクサ４１０と、フリップフロップ（ＦＦ：Flip Flop）４２０と、処理対象の１ビットについてパンクチャ／レピテションの有無を決定する演算部４３０とを備えている。

【００２６】

図４において演算部４３０は、レートマッチング率が０～３の範囲の処理に対応している。ここでレートマッチング率とは、入力ビット数に対する出力ビット数の割合を示す値である。言い換えると、演算部４３０は、対象ビットについて、パンクチャと、最大２回までのレピテションとを実行することができる（例えば、処理対象の１ビットを２回レピテションする場合は計３ビット出力されるため、レートマッチング率は３となる）。

【００２７】

演算部４３０は、パンクチャ・モードの場合は、対象ビットをパンクチャするときは、出力１から０を出力し、パンクチャしないときは、出力１から１を出力する（パンクチャ・モードの場合、出力２の値は不定）。

【００２８】

一方、レピテション・モードの場合は、対象ビットを１回レピテションするときは（出力１,出力２）＝（０,１）を出力し、２回レピテションするときは（出力１,出力２）＝（０,０）を出力する。レピテションしないときは、（出力１,出力２）＝（１,１）を出力する。

【００２９】

回路４００は、ＦＦ４２０にクロック（ＣＬＫ）が入力される度に処理対象の１ビットについてパンクチャ／レピテションの有無（レピテションの場合はその回数）を示す出力

１、出力２を演算部４３０から出力する。ＦＦ４２０には、対象ビットの直前ビットのパラメータｅが記憶されており、ＦＦ４２０は、クロックが入力されると、そのパラメータｅを加算部４３２へ出力する。加算部４３２は、ＦＦ４２０から入力されたｅとe_{minus}との差分値（$e-e_{minus}$）を計算し、判定部４３４と加算部４３６とへ当該差分値を出力する。

【００３０】

判定部４３４は、加算部４３２から入力された差分値（$e-e_{minus}$）が正であるか否かを判定し、正の場合は１を、０以下の場合は０を、出力１として外部へ出力するとともに、同じ値をマルチプレクサ４３８とＯＲ演算部４４０とへ出力する。マルチプレクサ４３８は、判定部４３４から入力された値が１の場合は０を、０の場合はe_{plus}を、加算部４３６へ出力する。

【００３１】

加算部４３６は、加算部４３２から入力された差分値（$e-e_{minus}$）と、マルチプレクサ４３８から入力された値とを加算して、判定部４４２と加算部４４４とへ出力する。このようにして、加算部４３６は、（$e-e_{minus}$）が０以下の場合にe_{plus}を加算する処理を行う。

【００３２】

判定部４４２、加算部４４４、マルチプレクサ４４６の処理は、判定部４３４、加算部４３６、マルチプレクサ４３８と同様である。図４のように、これらの要素は、加算部４３６がパラメータｅに対してe_{plus}を加算してもなお０以下の場合に、加算部４４４がe_{plus}をさらに加算するように構成されている。レピテション・モードにおいて、加算部４４４がe_{plus}を加算するのは、レピテションを２回行う場合に該当する。

【００３３】

なお、加算部４４４の出力はマルチプレクサ４１０へ入力される。マルチプレクサ４１０は、動作開始時にはe_{ini}をＦＦ４２０へ出力し、それ以降は加算部４４４からの出力をＦＦ４２０へ出力するように動作する。

【００３４】

このようにして、ＦＦ４２０にクロックが入力される度に、図２の処理が１ビット分実行されて、パンクチャの有無／レピテションの回数を示す出力１、出力２が演算部４３０から出力される。

【００３５】

図４を参照して説明した上記の構成では、パンクチャの有無／レピテションの回数を決定できるのは１クロックにつき入力データの１ビット分だけであり、しかもあるビットのパラメータｅはその直前のビットのパラメータｅに基づいて算出されるため入力データの先頭ビットから逐次的に処理を進めなければならない。このため、この構成では、レートマッチング／デレートマッチングの処理に要する時間が入力データのデータ長に比例し、十分な処理速度を得ることができない。

【００３６】

そこで、例えば、図４に示した演算部４３０を図５のように複数接続して、複数ビット分のパンクチャ／レピテションの有無を１クロックで決定する回路を構成することも考えられる。図５は、２つの演算部４３０が接続された回路５００の構成例を示す図である。回路５００は、ＦＦ４２０にクロックが入力される度に、図２の処理が２ビット分実行されるため、１クロックにつきパンクチャの有無／レピテションの回数を２ビットずつ決定することができる。

【００３７】

しかし、図５のように演算部３３０を複数接続すると、接続数に比例してクリティカルパスが長くなり、それに応じてロジック遅延が大きくなってしまう。これは、あるビットのパラメータｅがその直前ビットのパラメータｅに基づいて算出されるためである。ロジック遅延が大きいとクロック周波数が低く制限されるため、このアプローチによる処理の高速化にも限界がある。

【００３８】
そこで、本実施形態では、入力データをレートマッチングする場合に、この入力データに含まれる複数ビットのそれぞれ出力回数を、他のビットに対する処理の結果に関係なく、基本パラメータの初期値e_{ini}、減分値e_{minus}、増分値e_{plus}と、対象ビットの入力データにおける順番とのみに基づいて決定することで、高速に処理を行うことが可能な構成について説明する。
【００３９】
（レートマッチング処理１）
まず、レートマッチング処理１について、図６を参照して説明する。図６は、レートマッチング処理１の手順を示す図である。図１のアルゴリズムについて検討すると、対象ビットをパンクチャ又はレピテションするときは、パラメータｅに対して増分値e_{plus}が加算され、逆にパラメータｅに対して増分値e_{plus}が加算されたときは、対象ビットがパンクチャ又はレピテションされる。レートマッチング処理１ではこのことに着目し、入力データのｍ番目のビットについて、初期値e_{ini}から減分値e_{minus}をｍ回引き算した値を正（１以上の整数）にするために、当該値に対して加算すべき増分値e_{plus}の個数の最小値ｑ(ｍ)を計算し、ｑ(ｍ)－ｑ(ｍ－１)に基づいて当該ｍ番目のビットのパンクチャ／レピテションの判定を行う。
【００４０】
図６のように、レートマッチング処理１においては、パンクチャ・モードでは、１≦ｍ≦X_iを満たす整数ｍのそれぞれについて（X_i：入力データ長）、ｑ(ｍ－１)＜ｑ(ｍ)が成り立つならば、入力データのｍ番目のビット$x_{i,m}$をパンクチャする。これは（ｍ－１）番目のビットに対応するパラメータｅと、ｍ番目のビットに対応するパラメータｅとの間で、増分値e_{plus}が加算されているためである。ただし、
［式１］

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

である。ここで、e_{ini}、e_{minus}、e_{plus}は（背景技術）で説明したパラメータである。
【００４１】
一方、レピテション・モードでは、１≦ｍ≦X_iを満たす整数ｍのそれぞれについて、入力データのｍ番目のビット$x_{i,m}$を（ｑ(ｍ)－ｑ(ｍ－１)）回レピテションする。これは（ｍ－１）番目のビットに対応するパラメータｅと、ｍ番目のビットに対応するパラメータｅとの間で、増分値e_{plus}が加算されているならば、その加算回数だけレピテションされているためである。
【００４２】
上記のように、レートマッチング処理１では、入力データのｍ番目のビット$x_{i,m}$のパンクチャの有無、又は、レピテションの回数を決定するための処理は、他のビット$x_{i,k}$（ｋ≠ｍ）についての処理結果に依存しない。このため、レートマッチング処理１によれば、図７のように入力データの各ビットについての処理をそれぞれ別のプロセッサで並列的に実行することができる。
【００４３】
式１について、図１７を参照してさらに説明する。図１のように、レートマッチング・アルゴリズムにおいては、パンクチャ・モードとレピテション・モードのいずれの場合においても、ビット毎に「ｅ＝ｅ－e_{minus}」の処理を行う。このため、パラメータｅは、初期値e_{ini}が代入された後（１７０１）、入力データのｍ番目のビット処理が終了するまでに、減分値e_{minus}がｍ回差し引かれる（１７０２）ことになる。一方、図１のように、レートマッチング・アルゴリズムにおいては、パラメータｅから減分値e_{minus}を減算した結果、パラメータｅの値が０以下になった場合は正の値（１以上）になるまでパラメータｅに増分値e_{plus}を加算する。このため、ｍ番目のビット処理が終了するまでに、増分値e_{plus}がｑ(ｍ)回加算されて、ｍ番目のビット処理終了後のパラメータｅの値は正

（１以上の整数）となる（１７０３）。

【００４４】

図１７のように、ｍ番目のビット処理終了後のパラメータｅの値が正（１以上）となるためには、増分値e_{plus}をｑ(ｍ)回、加算した加算値が、１－（－ｍ・e_{minus}＋e_{ini}）＝ｍ・e_{minus}－e_{ini}＋１以上となる必要がある。従って、この条件を満たす加算回数ｑ(ｍ)は、ｍ・e_{minus}－e_{ini}＋１をe_{plus}で割って切り上げた値（式１）に等しい（ｍ・e_{minus}－e_{ini}＋１が負の場合は、$1 \leqq e_{ini} \leqq e_{plus}$よりｑ（ｍ)＝０が出力される）。

【００４５】

なお、$1 \leqq e_{ini} \leqq e_{plus}$であるため、ｑ(ｍ)は、以下のように変形できる。

［式２］

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil =-\left\lfloor \frac{e_{ini}-m\cdot e_{minus}-1}{e_{plus}} \right\rfloor$$

$$=\left\lceil \frac{m\cdot e_{minus}-e_{ini}}{e_{plus}}+1 \right\rceil =-\left\lfloor \frac{e_{ini}-m\cdot e_{minus}}{e_{plus}}-1 \right\rfloor$$

$$=\left\lceil \frac{m\cdot e_{minus}-e_{ini}}{e_{plus}} \right\rceil +1=-\left\lfloor \frac{e_{ini}-m\cdot e_{minus}}{e_{plus}} \right\rfloor +1$$

（レートマッチング処理２）

レートマッチング処理１のパンクチャリング・モードではｑ(ｍ－１)＜ｑ(ｍ)が成り立つ場合にビット$x_{i,m}$をパンクチャする。ここで、パンクチャリング・モードでは、同一ビットについて２回以上e_{plus}が加えられることはないため、ｑ(ｍ)－ｑ(ｍ－１)∈{０，１}である。従って、パンクチャリング・モードに基づいて入力データをレートマッチングすることは、元のビットも含めてビット$x_{i,m}$を{１－（ｑ(ｍ)－ｑ(ｍ－１)）}回出力することと等価である。

【００４６】

また、レートマッチング処理１のレピテション・モードでは、ビット$x_{i,m}$を（ｑ(ｍ)－ｑ(ｍ－１)）回レピテションするが、これは元のビットも含めてビット$x_{i,m}$を{ｑ(ｍ)－ｑ(ｍ－１)＋１}回出力することと等価である。

【００４７】

以上より、レートマッチング処理１（図６）は、

［式３］

$$g(m)=\begin{cases} 1-(q(m)-q(m-1)) & \text{（パンクチャ・モード）} \\ q(m)-q(m-1)+1 & \text{（レピテション・モード）} \end{cases}$$

を用いて図８で示されるレートマッチング処理２と等価である。図８はレートマッチング処理２の手順を示す図である。

【００４８】

レートマッチング処理２では、パンクチャ・モードとレピテション・モードの両方において、$1 \leqq m \leqq X_i$を満たす整数ｍのそれぞれについて（X_i：入力データ長）、入力データのｍ番目のビット$x_{i,m}$をｇ(ｍ)回出力する。従って、レートマッチング処理２によれば、入力データのビットと処理済みデータのビットとの関係を模式的に示すと、図９のようになる。

【００４９】

上記のように、レートマッチング処理２では、入力データのｍ番目のビット$x_{i,m}$に対応する出力ビット数は、他のビット$x_{i,k}$（ｋ≠ｍ）についての処理結果に依存しない。このため、レートマッチング処理２によれば、入力データの各ビットについての処理をそれぞれ別のプロセッサで並列的に実行することができる。

【００５０】

上記の出力回数ｇ(ｍ)を計算する回路は、例えば、図１０のような回路構成によって実行することができる。図１０は、出力回数ｇ(ｍ)を計算する回路１０００の一構成例を示す図である。図１０では、出力回数ｇ(ｍ)を連続する８ビットについて順次並列的に算出する回路が例示されている。

【００５１】

図１０のように、回路１０００は、演算部１００１～１００９、１０１１～１０１９、加算部１０２１～１０２８を備えている。演算部１００１～１００９は、ｎ＝０，１，２，・・・に対して、それぞれ８ｎ（１００１），８ｎ＋１（１００２），８ｎ＋２（１００３），・・・，８ｎ＋８（１００９）を計算し、それぞれ演算部１０１１～１０１９へ出力する。演算部１００１～１００９は、演算部１０１１～１０１９のそれぞれに対して、処理すべき入力データのビットの入力データにおける順番を割り当てている。

【００５２】

演算部１０１１～１０１９は、演算部１００１～１００９において計算されたｍ＝８ｎ，８ｎ＋１，８ｎ＋２，・・・，８ｎ＋８をそれぞれ入力して、

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

を計算する。そして、演算部１０１１は、ｑ(８ｎ)を加算部１０２１へ出力する。演算部１０１２～１０１８は、それぞれ計算結果を加算部１０２１及び１０２２、・・・、加算部１０２７及び１０２８へ出力する。演算部１００９は、ｑ(８ｎ＋８)を加算部１０２８へ出力する。

【００５３】

加算部１０２１～１０２８は、演算部１０１１～１０１９からの入力に基づいてそれぞれｇ(８ｎ＋１)，ｇ(８ｎ＋２)，・・・，ｇ(８ｎ＋８)を計算する。具体的には、ｍ＝８ｎ＋１，８ｎ＋２，・・・，８ｎ＋８に対して、パンクチャ・モードで動作する場合は、ｇ(ｍ)＝１－（ｑ(ｍ)－ｑ(ｍ－１)）を計算し、レピテション・モードで動作する場合は、ｇ(ｍ)＝ｑ(ｍ)－ｑ(ｍ－１)＋１を計算する。

【００５４】

上記のように、回路１０００は、入力データのｍ番目のビットを出力する出力回数ｇ(ｍ)を、入力データのそれぞれ異なるビットについて、該入力データの伸縮を行うための基本パラメータ初期値e_{ini}、減分値e_{minus}、増分値e_{plus}と、該ｍの値と、に基づいてそれぞれ決定する複数の決定手段を備えている。従って、当該構成によれば、複数ビットの出力回数を他のビットの処理に依存することなく並列的に決定することができる。

【００５５】

なお、図１０の例では、例えば、ｇ(８ｎ＋１)を決定するための決定手段は、演算部１０１１、１０１２、加算部１０２１により実現されており、ｇ(８ｎ＋２)を決定するための決定手段は、演算部１０１２、１０１３、加算部１０２２により実現されている。このように、図１０の例では、隣接する決定手段のｑ(ｍ)を決定するための演算部が共用されている。しかし、ｑ(ｍ)を決定するための演算部は決定手段ごとに別個に設けるようにしても構わない。例えば、加算部１０２１に入力されるｑ(８ｎ＋１)を計算するための演算部と、加算部１０２２に入力されるｑ(８ｎ＋１)を計算するための演算部とを別に設けてもよい。

【００５６】

なお、図１０の例では、出力回数ｇ(ｍ)を８ビットずつ決定する構成を例示したが、出力回数を決定するビット数は８個に限られず、任意の値にすることができる。

【００５７】

（ビットアドレス生成器）

{ｇ(１)，ｇ(２)，・・・，ｇ(X_i)}から所望のｇ(ｘ)（ｘ∈{１，２，・・・，X_i}）の値を参照する場合、図１１のように入力データのｍビット目に対するレートマッチング後のビット集合（ｇ(ｍ)個）の先頭ビットアドレス（先頭ビット位置）ａ(ｍ)が分かると、回

路設計上便利である。そこで、１≦ｍ≦X_iを満たす整数ｍのそれぞれについて（X_i：入力データ長）、先頭ビットのアドレスａ(ｍ)を計算し、ｇ(ｍ)に加えてａ(ｍ)をも出力する処理について説明する。なお、先頭ビットアドレスａ(ｍ)は、１から開始する値（ａ(ｍ)≧１）である。

【００５８】

まず、ビットアドレスａ(ｍ)を求めるための計算式を導出する。図１１のように、

ａ(ｍ)＝ａ(ｍ－１)＋ｇ(ｍ－１)

が成り立つ。従って、

$$a(m)=a(1)+\sum_{n=1}^{m-1}g(n)$$

が成り立つ。ｇ(ｍ)は［式３］のように与えられるから、ａ(１)＝１とすると、パンクチャリング・モードにおけるａ(ｍ)は以下の［式４］で与えられる。

［式４］

$$\begin{aligned}a(m)&=a(1)+\sum_{n=1}^{m-1}g(n)\\&=a(1)+\sum_{n=1}^{m-1}\{1-(q(n)-q(n-1))\}\\&=a(1)+(m-1)-q(m-1)+q(0)\\&=m-q(m-1)\end{aligned}$$

なお、［式１］より

$$q(0)=\left\lceil\frac{-e_{ini}+1}{e_{plus}}\right\rceil$$

であるが、（背景技術）で説明したように、１≦e_{ini}≦e_{plus}であるためｑ(０)＝０である。

【００５９】

一方、レピテション・モードにおけるａ(ｍ)は以下の［式５］で与えられる。

［式５］

$$\begin{aligned}a(m)&=a(1)+\sum_{n=1}^{m-1}g(n)\\&=a(1)+\sum_{n=1}^{m-1}\{(q(n)-q(n-1))+1\}\\&=a(1)+(m-1)+q(m-1)-q(0)\\&=m+q(m-1)\end{aligned}$$

レートマッチング処理３は、例えば、図１２のような回路構成によって実行することができる。図１２は、レートマッチング処理３を実行する回路（以下、ビットアドレス生成器）１２００の一構成例を示す図である。図１２では、図１０と同様にｇ(ｍ)を連続する８ビットについて順次並列的に算出する回路が例示されている。なお、図１０と同様の構成については同じ符号を付し、説明を省略する。

【００６０】

図１２のように、ビットアドレス生成器１２００は、図１０の回路１０００の構成要素に加えて、加算部１２０１～１２０８を備える。また、演算部１００２～１００９は計算結果（８ｎ＋１，・・・，８ｎ＋８）をそれぞれ加算部１２０１～１２０８へ出力するとともに、演算部１０１１～１０１９は計算結果（ｑ(８ｎ)，・・・，ｑ(８ｎ＋８)）をそれぞれ加算部１２０１～１２０８へ出力する。加算部１２０１～１２０８は、演算部１０

０２～１００９からの入力、及び、演算部１０１１～１０１８からの入力に基づいて、それぞれａ(８ｎ＋１),ａ(８ｎ＋２),・・・,ａ(８ｎ＋８)を計算する。具体的には、ｍ＝８ｎ＋１,８ｎ＋２,・・・,８ｎ＋８に対して、パンクチャ・モードで動作する場合は、ａ(ｍ)＝ｍ－ｑ(ｍ－１)を計算し、レピテション・モードで動作する場合は、ａ(ｍ)＝ｍ＋ｑ(ｍ－１)を計算する。

【００６１】

上記のようにビットアドレス生成器１２００は、入力データのｍ番目のビットについての、レートマッチング後の出力結果が、伸縮済みデータ（レートマッチング済みのデータ）において占めるべき箇所の先頭ビットアドレスａ(ｍ)を、基本パラメータ（初期値e_{ini}、減分値e_{minus}、増分値e_{plus}）と、該ｍの値とのみに基づいて決定する。このように、ａ(ｍ)を決定する際に他のビットに対する処理結果を利用しないため、ビットアドレス生成器１２００を用いることで入力データの任意のビット位置からレートマッチング／デレートマッチング処理を行うことができる。また、レートマッチング／デレートマッチング処理を複数のプロセッサ又はハードウエアで分散させて実行した場合、その処理結果を書き込むべきアドレスをビットアドレス生成器１２００を用いて簡易に求めることができる。

【００６２】

なお、図１１から明らかなように、入力データに対してレートマッチング処理を行い、レートマッチング後のデータをアドレスが１から始まるメモリに格納することは、入力データのｍビット目をメモリアドレスａ(ｍ)～{ａ(ｍ)＋ｇ(ｍ)－１}に格納することと等価である。図１３はこのようなレートマッチング処理（レートマッチング処理３）の手順を示している。

【００６３】

（レートマッチング装置）

次に、上述のビットアドレス生成器１２００を用いたレートマッチング装置の構成例について説明する。

【００６４】

図１４は、ビットアドレス生成器１２００を用いたレートマッチング装置１４００の一例を示す図である。レートマッチング装置１４００は、入力データについて８ビットずつ並列的にレートマッチングを行う。図１４のように、レートマッチング装置１４００は、アドレス生成器１２００、７個の加算部１４０１～１４０７、８個のビット位置デコーダ１４１１～１４１８、マスク１４２１～１４２８、ＯＲ演算部１４３０を備える。

【００６５】

ビットアドレス生成器１２００は、カウンタ（不図示）からｎ＝０，１，２，・・・を入力し、入力データの（８ｎ＋１）～（８ｎ＋８）番目のビットについての、伸縮済みデータにおいて占めるべき箇所の先頭ビットアドレスａ(８ｎ＋１)，・・・，ａ(８ｎ＋８)と、出力回数ｇ(８ｎ＋１)，・・・，ｇ(８ｎ＋８)を決定する。レートマッチング装置１４００は、これらの先頭ビットアドレス、出力回数に基づいて入力データの（８ｎ＋１）～（８ｎ＋８）番目のビット値ｄ(８ｎ＋１)，・・・，ｄ(８ｎ＋８)を、それぞれビットアドレスａ(８ｎ＋１)～{ａ(８ｎ＋１)＋ｇ(８ｎ＋１)－１}、・・・、ａ(８ｎ＋８)～{ａ(８ｎ＋８)＋ｇ(８ｎ＋８)－１}へ出力するための処理を行う。

【００６６】

ビットアドレス生成器１２００において決定されたビットアドレスａ(８ｎ＋１)は、加算器１４０１～１４０７へ出力され、ビットアドレスａ(８ｎ＋２)～ａ(８ｎ＋８)は加算器１４０１～１４０７へそれぞれ出力される。また、出力回数ｇ(８ｎ＋１)～ｇ(８ｎ＋８)は、それぞれビット位置デコーダ１４１１～１４１８へ出力される。

【００６７】

加算器１４０１～１４０７は、アドレスａ(８ｎ＋２)，・・・，ａ(８ｎ＋８)からそれぞれａ(８ｎ＋１)を減算し、当該減算結果をオフセット値としてビット位置デコーダ１４１２～１４１８へ出力する。

【００６８】
ビット位置デコーダ１４１１～１４１８は、それぞれ伸縮済みデータにおいて入力データの（８ｎ＋１）～（８ｎ＋８）番目のビットが占めるべきビット位置にビット１を、それ以外のビット位置にビット０を出力する。具体的には、加算器１４０１～１４０７から入力されたオフセット値の個数のビット０を出力し、続けてアドレス生成器１２００から入力されたビット出力回数個のビット１を出力する。そして、残りのビット位置にはビット０を出力する。ただし、ビット位置デコーダ１４１１は、一定のオフセット値として０が入力される。各ビット位置デコーダ１４１１～の出力は対応するマスク１４２１～１４２８へ入力される。
【００６９】
例えば、図１４では、ｇ(８ｎ＋１)＝２、ｇ(８ｎ＋２)＝２、ｇ(８ｎ＋８)＝１、ａ(８ｎ＋２)＝３、ａ(８ｎ＋８)＝１３の場合が例示されている。この場合、ビット位置デコーダ１４１１に入力されるオフセット値は０、ビット出力回数は２（＝ｇ(８ｎ＋１)）であるため、ビット位置デコーダ１４１１は、ビット１を続けて２個出力し、残りのビット位置にはビット０を出力している。ビット位置デコーダ１４１２に入力されるオフセット値は２（＝ａ(８ｎ＋２)－ａ(８ｎ＋１)）、ビット出力回数は２（＝ｇ(８ｎ＋２)）であるため、ビット位置デコーダ１４１２は、ビット０を続けて２個出力し、ビット１を続けて２個出力し、残りのビット位置にはビット０を出力している。ビット位置デコーダ１４１８に入力されるオフセット値は１２（＝ａ(８ｎ＋８)－ａ(８ｎ＋１)）、ビット出力回数は１（＝ｇ(８ｎ＋８)）であるため、ビット位置デコーダ１４１８は、ビット０を続けて１２個出力し、ビット１を１個出力している。
【００７０】
マスク１４２１～１４２８は、入力データが記憶された記憶装置（ＲＡＭ、レジスタ等）１４５０から、（８ｎ＋１）～（８ｎ＋８）番目のビット値ｄ(８ｎ＋１)，・・・，ｄ(８ｎ＋８)をそれぞれ読み出し、ビット位置デコーダ１４１１～１４１８からの出力とのマスクをとって、ＯＲ演算部１４３０へ出力する。即ち、各マスク１４２１～１４２８は、記憶装置１４５０から読み出した入力データの対象ビットを、ビット位置デコーダからビット１が出力された位置に出力し、それ以外の位置にはビット０を出力する。
【００７１】
これにより、マスク１４２１～１４２８は、それぞれ入力データのビット値ｄ(８ｎ＋１)，・・・，ｄ(８ｎ＋８)を、それぞれビットアドレスａ(８ｎ＋１)～{ａ(８ｎ＋１)＋ｇ(８ｎ＋１)－１}、・・・、ａ(８ｎ＋８)～{ａ(８ｎ＋８)＋ｇ(８ｎ＋８)－１}へ出力し、それ以外のビットアドレスにはそれぞれビット０を出力することになる。
【００７２】
ＯＲ演算部１４３０は、マスク１４２１～１４２８からの出力ビットを、ビット位置毎にＯＲ演算して、出力する。従って、入力データの（８ｎ＋１）～（８ｎ＋８）番目のビット値ｄ(８ｎ＋１)，・・・，ｄ(８ｎ＋８)が、それぞれビットアドレスａ(８ｎ＋１)～{ａ(８ｎ＋１)＋ｇ(８ｎ＋１)－１}、・・・、ａ(８ｎ＋８)～{ａ(８ｎ＋８)＋ｇ(８ｎ＋８)－１}へ出力される。
【００７３】
上記のように、レートマッチング装置１４００は、入力データの複数ビットを、当該ビットについてビットアドレス生成器１２００が出力する出力回数並列的に出力する。このため、レートマッチング装置１４００は、ある処理対象ビットについてパンクチャリング／レピテションを決定するために、先頭ビットからその処理対象ビットまで逐次的にパラメータｅを算出する必要がある従来の構成と比べて高速にレートマッチング処理を実行することが可能である。
【００７４】
なお、本実施形態では、レートマッチングされた伸縮済みデータは、図１５に示すような１アドレスブロックにＮビットのデータが格納可能なＲＡＭ１４７０に、ＲＡＭアドレス０のブロックの先頭から順に格納される。このため、ＯＲ演算部１４３０から出力され

るレートマッチング結果は、図１４のように、ＲＡＭアドレスが

$$\left\lfloor \frac{a(8n+1)-1}{N} \right\rfloor$$

のブロックの、ビット位置（ａ(８ｎ＋１)－１）ｍｏｄ Ｎから格納される。

【００７５】

（デレートマッチング装置）

次に、上述のビットアドレス生成器１２００を用いたデレートマッチング装置の構成例について説明する。受信データの誤り訂正復号を行う場合、受信側は、受信データの軟判定値列をもとに、送信データが候補のビット列と一致する確率を計算して、復号後のデータの候補を絞りこんでいく。ここで、送信データと候補のビット列とが一致する確率の計算は、送信データの各ビットに割り当てられたエネルギー（送信電力）に基づいて行われる。このため、誤り訂正復号を適切に行うためには、送信データの各ビットに割り当てられたエネルギーを求める必要がある。

【００７６】

レートマッチング後のデータは、１ビットあたり一定のエネルギーで送信されるため、一般に、レートマッチング後のビット数（重み）が多いほど、レートマッチング前の対応するシンボルに割り当てられるエネルギーが大きくなる。このため、本実施形態では、受信データ（伸縮前データ）のビットの重み付き軟判定値を出力する。これにより、各ビットに割り当てられたビット重みに対応するエネルギーを推定することができ、誤り訂正復号を適切に行うことが可能である。

【００７７】

図１６は、ビットアドレス生成器１２００を用いたデレートマッチング装置１６００の一例を示す図である。デレートマッチング装置１６００は、レートマッチング処理により伸縮された伸縮データについて８ビットずつ並列的にデレートマッチングを行う。図１６のように、デレートマッチング装置１６００は、アドレス生成器１２００、７個の加算部１４０１～１４０７、８個のビット位置デコーダ１４１１～１４１８、８個の軟判定値選択器１６０１～１６０８、８個の加算器１６１１～１６１８を備える。アドレス生成器１２００、加算部１４０１～１４０７、ビット位置デコーダ１４１１～１４１８は、図１４を参照して説明したものと同様であるため、説明を省略する。なお、デレートマッチング装置１６００では、ビット位置デコーダ１４１１～１４１８の出力は、それぞれ軟判定値選択器１６０１～１６０８へ入力される。

【００７８】

ビットアドレス生成器１２００は、カウンタ（不図示）からｎ＝０，１，２，・・・を入力し、伸縮前データの（８ｎ＋１）～（８ｎ＋８）番目のビットについての、伸縮済みデータにおける対応箇所の先頭ビットアドレスａ(８ｎ＋１)，・・・，ａ(８ｎ＋８)と、出力回数ｇ(８ｎ＋１)，・・・，ｇ(８ｎ＋８)を決定する。デレートマッチング装置１４００は、これらの先頭ビットアドレスから開始するビットを用いて、伸縮前データの各ビットを出力する。より具体的には、各先頭ビットアドレスａ(８ｎ＋１)，・・・，ａ(８ｎ＋８)から、それぞれ出力回数ｇ(８ｎ＋１)，・・・，ｇ(８ｎ＋８)個分の伸縮済みデータのビットを読み出し、当該回数個分の伸縮済みデータのビットの重み付き平均値を、それぞれ伸縮前データのビットとして出力していく。

【００７９】

デレートマッチング前のデータ（伸縮前データ）は、図１５を参照して説明した、１アドレスブロックにＮビットのデータが格納可能なＲＡＭ１４７０に事前にプリフェッチされる。デレートマッチング前のデータも、ＲＡＭアドレス０のブロックの先頭から順に格納される。

【００８０】

不図示のカウンタよりｎ＝０，１，２，・・・が入力されると、レートでマッチング前のデータが、ＲＡＭアドレスが

$$\left\lfloor \frac{a(8n+1)-1}{N} \right\rfloor$$

のブロックの、ビット位置（ａ(８ｎ＋１)－１）ｍｏｄ Ｎから順に読み出されていき、各軟判定値選択器１６０１～１６０８へ入力される。これと同時に、アドレス生成器１２００は伸縮前データの（８ｎ＋１），・・・，（８ｎ＋８）番目のビットについての、出力回数ｇ(８ｎ＋１)，・・・，ｇ(８ｎ＋８)、先頭ビットアドレスａ(８ｎ＋１)，・・・，ａ(８ｎ＋８)を出力し、これらの値に基づいて、各ビット位置デコーダ１４１１～１４１８からａ(８ｎ＋１)，・・・，ａ(８ｎ＋８)に基づいてオフセットされた位置からそれぞれｇ(８ｎ＋１)，・・・，ｇ(８ｎ＋８)のビット１が出力される。そして、各軟判定値選択器１６０１～１６０８は、ビット位置デコーダからビット１が入力された位置に対応する軟判定値をそれぞれ加算器１６１１～１６１８へ出力する。

【００８１】

加算器１６１１～１６１８は、それぞれ入力された軟判定値の和（デレートマッチングの前のビット数で重み付けられた平均値）を計算し、デレートマッチング値として出力する。例えば、レピテションされて３ビットになったビットは、レピテションされて２ビットになったビットと比較して、デレートマッチング後の軟判定値は１．５倍の重みを持つ。また、パンクチャリングされたビットがデレートマッチングされるときは、重みが０となるため、軟判定値として０が出力される。

【００８２】

上記のように、デレートマッチング装置１６００は、伸縮前データの複数ビットを、当該ビットについてビットアドレス生成器１２００が出力する先頭ビットアドレスから開始するビットを用いて、伸縮済みデータから並列的に復元する。このため、デレートマッチング装置１６００は、先頭ビットから処理対象ビットまで逐次的にパラメータｅを算出する必要がある従来の構成と比べて高速にデレートマッチング処理を実行することが可能である。

【００８３】

また、デレートマッチング装置１６００は、ビットアドレス生成器１２００が出力した先頭ビットアドレスから、出力回数個分の伸縮済みデータのビットを読み出し、この回数個分の伸縮済みデータのビットの軟判定値の和を、伸縮前データのビットの重み付き軟判定値として出力する。このため、デレートマッチング装置１６００から出力された重み付き軟判定値に基づいて、各ビットに割り当てられたエネルギーを推定し、誤り訂正符号化された受信データを適切に復号することができる。

【００８４】

以上、本発明の実施形態例について詳述したが、本発明はレートマッチング装置、デレートマッチング装置の他、コンピュータに上記処理を実行させる方法、プログラムもしくは当該プログラムを記憶した記録媒体としての実施態様をとることが可能である。

【図面の簡単な説明】

【００８５】

【図１】３ＧＰＰリリース６のレートマッチング処理のアルゴリズムを示す図である。

【図２】レートマッチング処理の実行例を示す図である。

【図３】通信装置の機能構成の概略を示すブロック図である。

【図４】レートマッチングを行う回路の一例を示す図である。

【図５】２つの演算部が接続された回路の構成例を示す図である。

【図６】レートマッチング処理１の手順を示す図である。

【図７】レートマッチング処理を複数のプロセッサを用いて並列的に行う様子を示す図である。

【図８】レートマッチング処理２の手順を示す図である。

【図９】レートマッチング処理２における、入力データのビットと処理済みデータのビットとの関係を模式的に示す図である。

【図１０】出力回数ｇ(ｍ)を計算する回路の一構成例を示す図である。
【図１１】入力データのｍビット目に対するレートマッチング後のビット集合の先頭ビットアドレスａ(ｍ)を示す図である。
【図１２】ビットアドレス生成器の一構成例を示す図である。
【図１３】レートマッチング処理３の手順を示す図である。
【図１４】ビットアドレス生成器を用いたレートマッチング装置の一例を示す図である。
【図１５】１アドレスブロックにＮビットのデータが格納可能なＲＡＭの構成を模式的に示す図である。
【図１６】ビットアドレス生成器を用いたデレートマッチング装置の一例を示す図である。
【図１７】パラメータｅと増分値e_{plus}の加算回数ｑ（ｍ）の関係を示す図である。

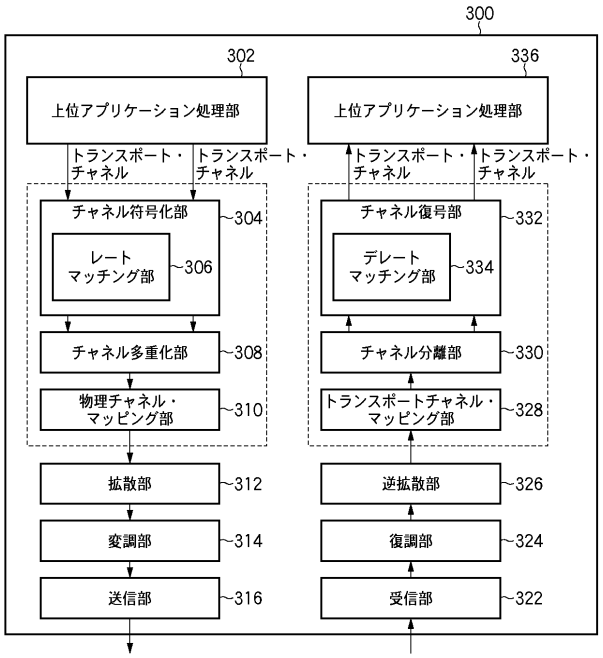
【図１】

```
if (パンクチャ実行) {
   e=e_ini                    --eの初期値
   m=1                        --カレントビットのインデックス
   while (m≦X_i) {
      e=e−e_minus             --eを更新
      if (e≦0) {              --mビット目をパンクチャするかチェック
         ビットx_i,mをパンクチャ
         e=e+e_plus           --eを更新
      }
      m=m+1                   --次のビットへ
   }
}
else {                        --レピテション実行
   e=e_ini                    --eの初期値
   m=1                        --カレントビットのインデックス
   while (m≦X_i) {
      e=e−e_minus             --eを更新
      while (e≦0) {           --mビット目をレピテションするかチェック
         ビットx_i,mをレピテション
         e=e+e_plus           --eを更新
      }
      m=m+1                   --次のビットへ
   }
}
```

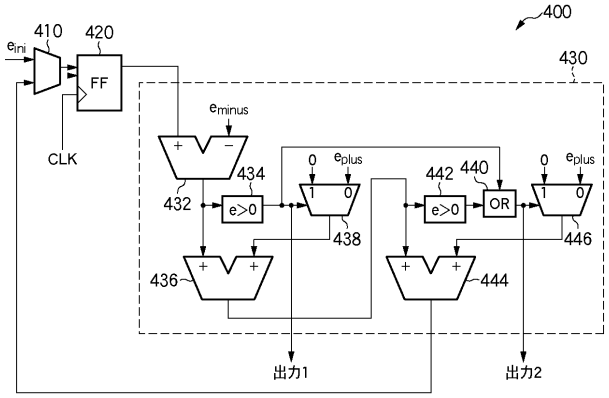
【図２】

m	e	m	e	m	e	m	e
1	−5	9	11	17	−5	25	11
2	21	10	5	18	21	26	5
3	15	11	−1	19	15	27	−1
4	9	12	25	20	9	28	25
5	3	13	19	21	3	29	19
6	−3	14	13	22	−3	30	13
7	23	15	7	23	23	31	7
8	17	16	1	24	17	32	1

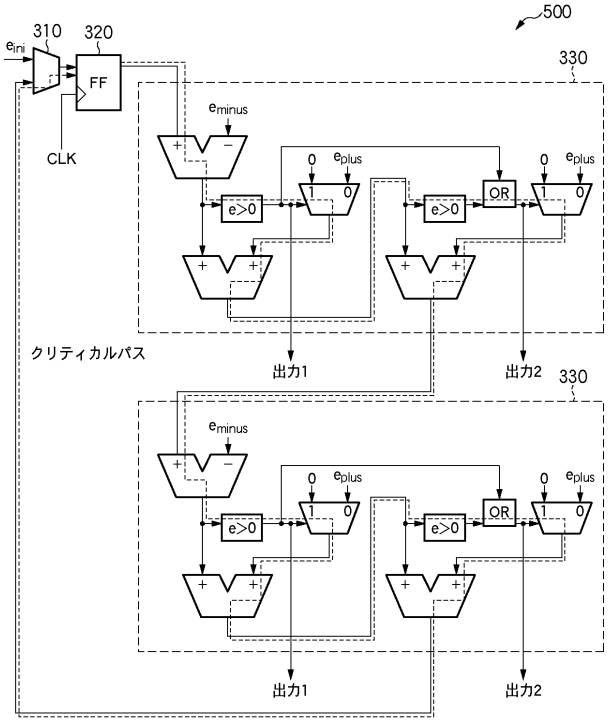
【図３】



【図４】



【図５】

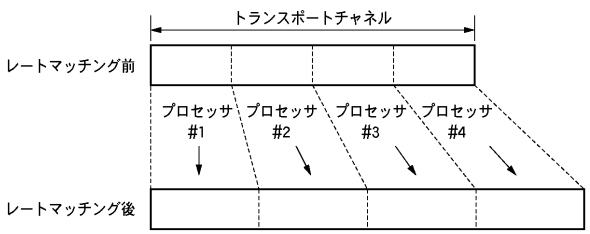


【図６】

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

```
if (パンクチャ実行) {
   m=1                          --カレントビットのインデックス
   while (m≦Xi) {
      if (q(m−1)<q(m)) {        --mビット目をパンクチャするかチェック
         ビットxi, mをパンクチャ
      }
      m=m+1                     --次のビットへ
   }
}
else {                          --レピテション実行
   m=1                          --カレントビットのインデックス
   while (m≦Xi) {
      ビットxi, mを(q(m)−q(m−1))回レピテション
      m=m+1                     --次のビットへ
   }
}
```

【図７】



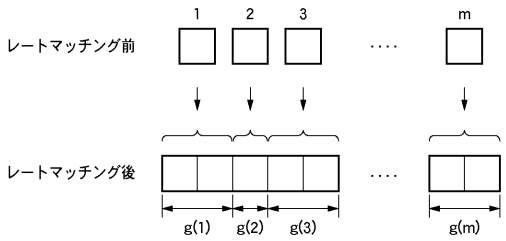
【図８】

$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

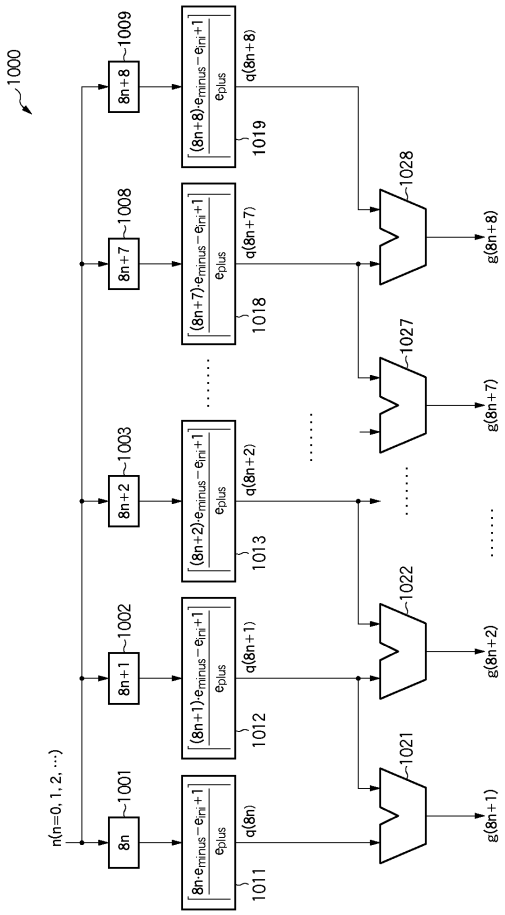
$$g(m)=\begin{cases} 1-(q(m)-q(m-1)) \text{ (パンクチャ・モード)} \\ q(m)-q(m-1)+1 \text{ (リピテション・モード)} \end{cases}$$

```
m=1                             --カレントビットのインデックス
while (m≦Xi) {
   ビットxi, mをg(m)回出力
   m=m+1                        --次のビットへ
}
```

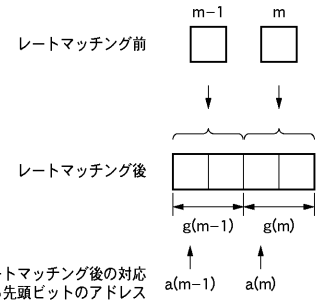
【図９】



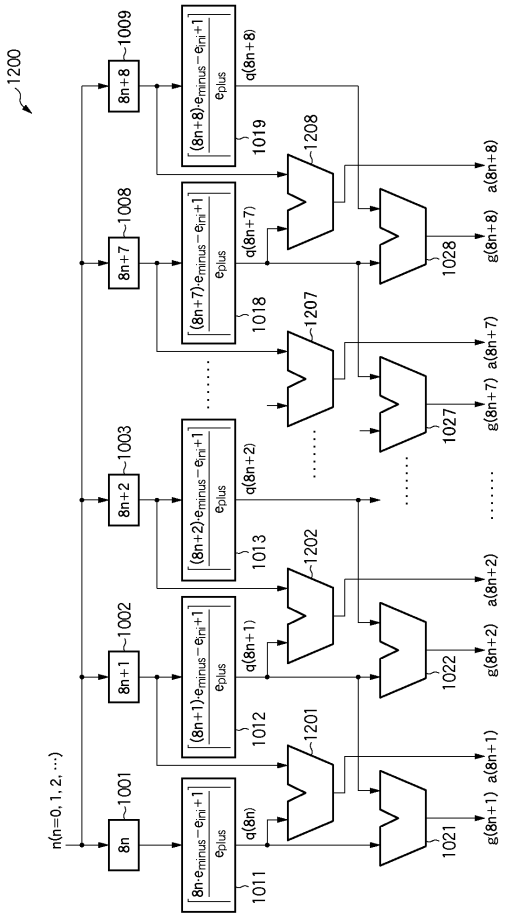
【図１０】



【図１１】



【図１２】



【図１３】

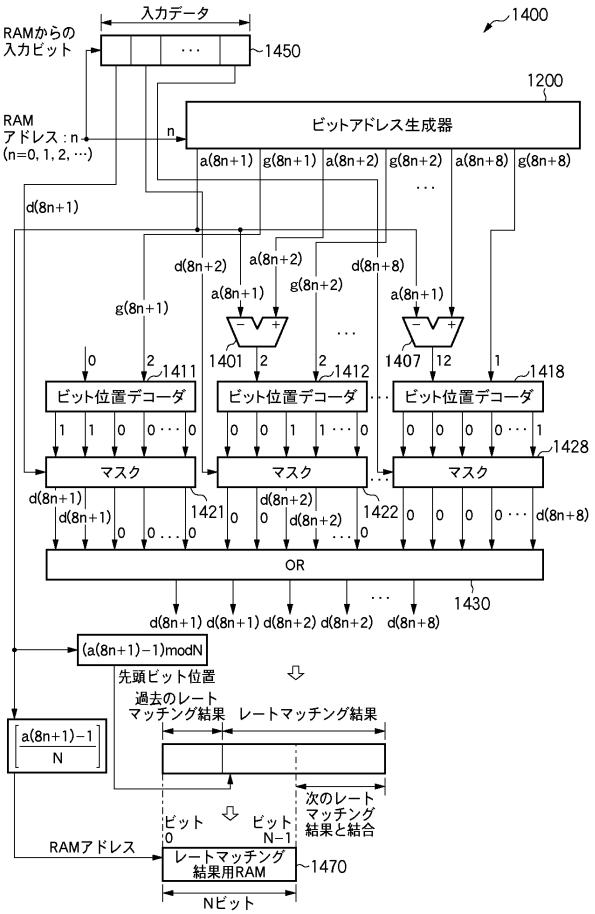
$$q(m)=\left\lceil \frac{m\cdot e_{minus}-e_{ini}+1}{e_{plus}} \right\rceil$$

$$g(m)=\begin{cases}1-(q(m)-q(m-1)) \text{ (パンクチャ・モード)}\\ q(m)-q(m-1)+1 \text{ (リピテション・モード)}\end{cases}$$

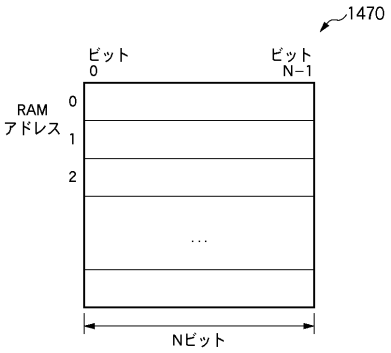
$$a(m)=\begin{cases}m-q(m-1) \text{ (パンクチャ・モード)}\\ m+q(m-1) \text{ (リピテション・モード)}\end{cases}$$

```
m=1                                  --カレントビットのインデックス
while (m≦Xi) {
  ビットxi, mをメモリアドレスa(m)～a(m)+g(m)−1へ格納
  m=m+1                              --次のビットへ
}
```

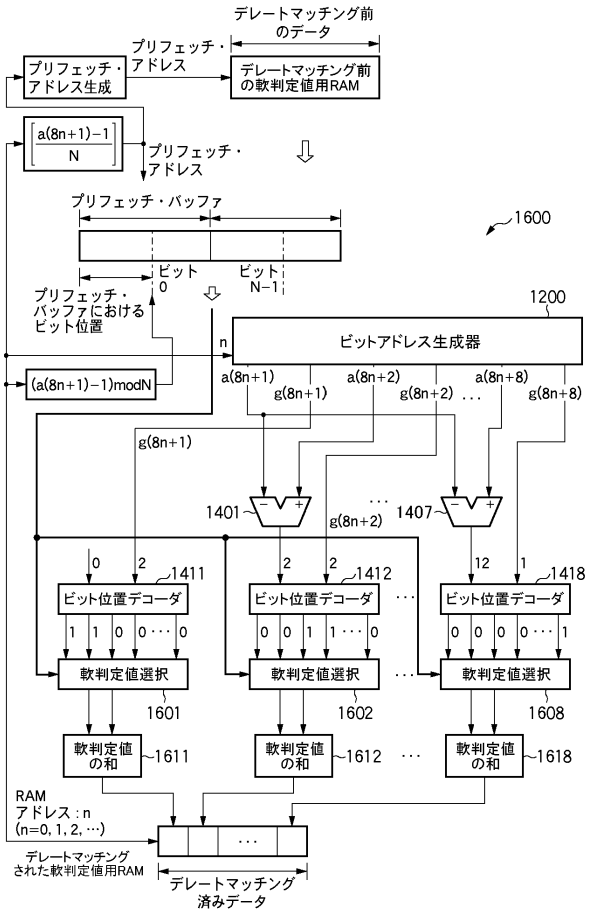
【図１４】



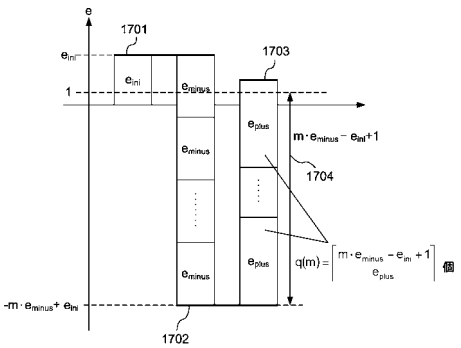
【図１５】



【図１６】



【図１７】



フロントページの続き

(72)発明者　浅中　和典
神奈川県横浜市港北区鳥山町１２６７－１０５

Ｆターム(参考) 5K022 EE02 EE14 EE21 EE31
5K067 AA21 BB04 BB21 CC10 DD27 DD51 EE02 EE10 HH22